Automatic Calibration of Large Traffic Models

Félix Mézière* Gabriel Gomes**
California PATH Program
University of California Berkeley

Je soussigné Félix Mézière certifie sur l'honneur :

- Que les résultats décrits dans ce rapport sont l'aboutissement de mon travail.
- Que je suis l'auteur de ce rapport.
- Que je n'ai pas utilisé des sources ou résultats tiers sans clairement les citer et les référencer selon les règles bibliographiques préconisées.

Je déclare que ce travail ne peut être suspecté de plagiat.

Date : samedi 23 août 2015

Signature :

*Ecole polytechnique [felix.meziere@polytechnique.edu]
**University of California Berkeley [gomes@path.berkeley.edu]

Abstract

We build a freeway traffic model calibration method for the input flows. The central feature is to make the model output match the location and times of the congestion. The main purpose of this work is to be able to reproduce the traffic behavior for a given weekday on a partially monitored freeway. An important constraint is that the non-monitored ramps flows should be as realistic as possible.

Index Terms

Large traffic model, CTM, calibration, imputation, optimization, black box, evolutionnary algorithm, CMA-ES.

Introduction

THIS paper solves a part of the large traffic model calibration problem. By calibrating such model, we mean that we try to reproduce as accurately as possible all the traffic phenomena on the freeway, while giving it an input as close to reality as possible (two goals than can be contradictory).
Large traffic models possess many input parameters, especially road characteristics and flow demands on each ramp. Our method partially solves the problem of the flow demands imputation.
On an incompletely monitored freeway scenario, we input to the monitored ramps the flows given by the measurements. We then associate to each non-monitored ramp a custom normalized flow profile called *template* and an intensity coefficient called *knob*. The input flow to the ramp will be the template multiplied by the knob. The method tries to match the mainline (i.e. main road, by opposition to ramps) measurements in terms of congestion location and times, as well as two other performance metrics. The variables the method can act on are only the values of the knobs.
The method is just the first-step in a wider context where the template shapes will also be modificable. Later, the calibration of other parameters (such as the *fundamental diagrams*, characteristics of the mainline links) will be in a loop with the flow calibration, in order to calibrate the model as a whole.
This method, although being very formalized in this paper, is simple and intuitive. This is the report on an empirical progressive construction.

I. Modeling and notation

A. Freeway model

The freeway model is composed by the physical characteristics of the freeway. We consider a one-way segment of freeway and its on-ramps and off-ramps.
We define here the components of the considered freeway model :

- *Mainline*: The freeway itself i.e. the central part where the cars go fast.
- *Ramps*: The portions of road connected to the mainline that allow to enter or exit it.
- *On-ramp*: Ramp to enter the mainline.
- *Off-ramp*: Ramp to exit the mainline.
- *Link*: a link is a segment of freeway or a ramp. The links are separated by *nodes*. Each ramp is connected to a node and a node can only be connected to one ramp. There can be several consecutive mainline links without ramps.
- *Linear order*: Links ordered accordingly to the traffic direction.
- *Number of lanes of a link*: number of cars than can be side by side on the same level of the link.
- *Topography of the freeway*: The length of the links and their number of lanes.
- *Source*: Link which is an on-ramp or the entry of the mainline.
- *Sink*: Link which is an off-ramp or the exit of the mainline.
- *Monitored link*: Link which possesses a fully functional sensor monitoring all of its lanes in terms of flow. If it is a mainline link, it must also be monitored in terms of density (or speed, which are equivalent).

B. Traffic model

The traffic model is plug into the freeway model. It is the set of rules that define how the traffic behaves on the freeway.

Definitions:

- *Scenario*: the freeway and traffic models and all the input of the traffic model.
- *Exit flow profile for a link*: Number of vehicles that actually exit the link during every time-step of a time profile.
- *Exit flow demand profile for a link*: exit flow profile for the link *inputted* to the traffic model. This can differ from the actual output flow profile if there are not enough cars on the freeway or if there is extreme congestion, in which case the cars cannot enter the freeway at the asked rhythm, forming a queue.
- *Density profile for a link*: average of the number of cars simultaneously on the link during every time-step. This can be expressed in cars or in cars/mile.

The *large traffic model* to calibrate is characterized by the following inputs and outputs:

- *Input:*
  - duration of the scenario and time step.
  - value of the exit flow demand at every source and off-ramp, for every time step (sum of the flow during sthe time step).
  - other parameters proper to the model, assumed to be already calibrated
- *Output:*
  - value of the exit flow on every link, for every time step (sum of the flow during the time step).
  - value of the density on every link, for every time step (average over the time step).

We make the following *assumptions* on the traffic model:

- The number of cars on the freeway at the beginning and the end of the time period is very small in comparison with the total number of cars going through the freeway during the period (for example, it is true if we take a period from midnight to midnight).
- There is approximately no queues on the ramps and the off-ramps can always obtain the flow their demand profile ask from the mainline.
  We justify this by the fact that the situations where this is not true are highly non-realistic and therefore far from our objective. There is no queue in the ramps because the flows we input them will always be inferior to the capacity of each ramp (physical box constraints). In addition, on the on-ramps, we have to make the additional usual hypothesis in traffic studies that the cars do not have difficulties to enter the mainline, even if there is congestion.

C. Data

We assume that we are in possession of the following measurements on the freeway:

- Value of the exit flow on some of the mainlines links, for every time step.
- Value of the density on the same mainline links, for every time step.
- Value of the exit flow on some of the ramps, for every time step.

Data assumptions:

- The first and last mainline link (entrance and exit of the freeway portion) are monitored.

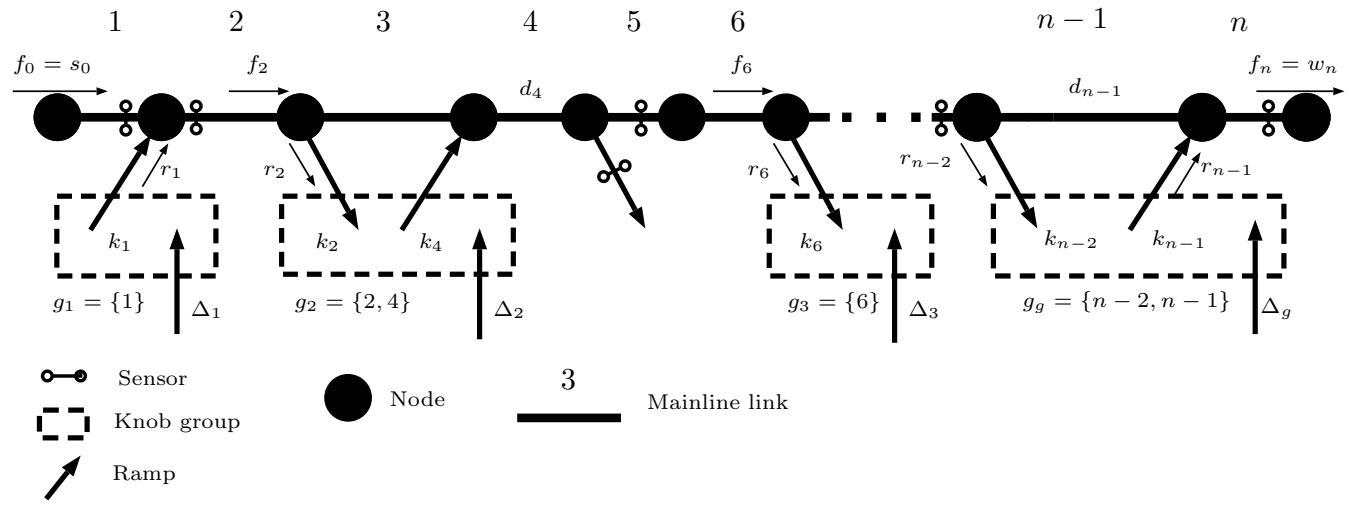


Fig. 1. Freeway model and notation

D. Notation

1) Freeway model: Let $M = [\![1, n]\!]$ the set of *mainline link* indexes and $R \subset M$ the set of mainline link indexes whose exit node is connected to a ramp (i.e. the set of ramps, indexed by their preceding mainline link).
We denote as $T \subset M$ the set of monitored mainline links (T for 'tracked') and $K = \{i_1, i_2, ..., i_\kappa\} \subset R$ the set of the κ non-monitored ramps (K for "knobs").
$(L_i)_{i \in M}$ are the lengths of the mainline links.

2) Traffic model: Let dt the time step of the model and D its duration. There is a total of $\frac{D}{dt} + 1$ time steps in the scenario. We will denote the set of these times steps $\tau = dt.[\![0, \frac{D}{dt}]\!]$
$\forall t \in \tau,\ (f_i(t))_{i \in M}$ and $(r_i(t))_{i \in R}$ are the flows exiting respectively the mainline and ramp links at time t, output by the model.
In addition, $f_0(t)$ is the flow entering the first mainline link (entrance of the freeway).
$\forall t \in \tau,\ (d_i(t))_{i \in M}$ are the densities on the mainline links, output by the model.
$\forall t \in \tau,\ \{\bar{f}_0(t); (\bar{r}_i(t))_{i \in M}\}$ are the exit flow demands inputted to the model.

3) Data: As a common pattern, the measured values will be denoted with a tilde.

Measured mainline exit flows are denoted $(\widetilde{f}_i(t))_{i \in T}$
Measured mainline densities are denoted $(\widetilde{d}_i(t))_{i \in T}$
Measured ramp exit flows are denoted $(\widetilde{r}_i(t))_{i \in (R \setminus K)}$

Fig. 1 summarizes the model. The missing notation will be introduced progressively.

II. Uncertainty

Our problem involves three sources of uncertainty or inexactness:

- Uncertainty on the data: the measurements have a certain confidence interval.
- Inexactness of the model itself.
  This inexactness reflects the fact that, even if we had perfect data and the demand on every ramp, the model would not output the exact real traffic (and congestion phenomena etc.).
- Inexactness of the shapes of the templates, that forces us to give freedom to the parameters.

These uncertainties and inexactness are merged into two uncertainties:

- *Uncertainty on the local flow measurements:* This describes the uncertainty at the link level. It is applied to the sum of all the flow measurements of one sensor during the duration.
  Denoting $F_i = \sum_{t\in\tau} f_i(t)$ and $\widetilde{F_i} = \sum_{t\in\tau} \widetilde{f}_i(t)$, this local uncertainty is divided into two competing components:
  - *additive local uncertainty*: denoted U^{add}. The additive confidence interval for F_i is:
  $$F_i \in [\widetilde{F_i} - U^{add},\ \widetilde{F_i} + U^{add}]$$
  - *multiplicative local uncertainty*: denoted U^{mul}. The multiplicative confidence interval for F_i is:
  $$F_i \in [\widetilde{F_i}.(1 - U^{mul}),\ \widetilde{F_i}.(1 + U^{mul})]$$
- *Uncertainty on the global duration-long measurements*: This describes the uncertainty at the whole mainline level. It is a generic multiplicative uncertainty applied to all quantities that are computed from the measurements on every mainline sensor and during the whole duration. We denote this uncertainty U^{global}.

  Let $\widetilde{q_i}(t)$ a quantity computed from the measurements on link i at time t. Denoting $Q = \sum_{i\in T}\sum_{t\in\tau} q_i(t)$ and $\widetilde{Q} = \sum_{i\in T}\sum_{t\in\tau} \widetilde{q_i}(t)$, the global confidence interval for this quantity is:
  $$Q \in [\widetilde{Q}.(1 - U^{global}),\ \widetilde{Q}.(1 + U^{global})]$$

The reader will understand better the form chosen for the uncertainty further on.

III. Problem formulation

A. Introduction

For every monitored source or off-ramp, we input to the model the measured flow as exit flow demand. The assumptions made in I-B imply that this demand is approximately equal to the actual model-output flow going through the ramp, for all times and above mentioned links:

$$\forall i \in R\backslash K,\ \forall t \in \tau,\ \widetilde{r_i}(t) = \bar{r_i}(t) \approx r_i(t) \tag{1}$$

Therefore, the only missing parameters to the model are the flow demand profiles of the non-monitored ramps: $(\bar{r_i}(t))_{\substack{i\in K\\ t\in\tau}}$. Our method consists in mapping these κ flow profiles into one parameter each.
To do that, a flow profile called *template* is built for every non-monitored ramp. These templates, denoted $(t_i(t))_{\substack{i\in K\\ t\in\tau}}$, consist in a normalized flow profile: a flow value is given to each element of τ and the resulting profile is normalized to a reasonable value Θ. For each of the non-monitored ramps i, we define a multiplicative factor k_i called *knob* that will set the intensity of the template. That is, we input as exit flow demand of the ramp its corresponding template multiplied by the ramp knob : $k_i.t_i(t)$.

The parameters of our imputation problem are therefore the κ **knobs**, corresponding to the κ non-monitored ramps.

In addition, due the same assumptions that gave Eq. 1, we have :

$\forall i \in K,\ \forall t \in \tau,\ k_i.t_i(t) = \bar{r_i}(t) \approx r_i(t)$

and, especially, with $\Theta = \sum_{t\in\tau} t_i(t)$:

$$\sum_{t\in\tau} r_i(t) = k_i.\Theta \tag{2}$$

B. Constraints on the parameters

We define here the constraints verified by the knobs. They consist in box hard boundaries and linear inequalities.

Notation:

$\vec{k} = (k_{i_1}, k_{i_2}, ..., k_{i_\kappa})$ is the vector containing the values of the knobs.
$\sigma = (\sigma_{i_1}, \sigma_{i_2}, ..., \sigma_{i_\kappa})$ is the source/sink indicator vector for the knobs:

$$\forall j \in K,\ \sigma_j = \begin{cases} 1\ if\ ramp\ j\ is\ an\ on\text{-}ramp \\ -1\ if\ ramp\ j\ is\ an\ off\text{-}ramp \end{cases}$$

For clarity, we will often abusively use the expression *knob-ramp i* instead of *ramp corresponding to the knob i*.

1) Physical boundaries: The box constraints applied to each knob are physical capacity limits imposed by the ramp they are associated with. They reflect that the maximum value of the ramp flow cannot exceed the capacity of the ramp.

$\forall i \in K$, the maximum m_i of knob i is defined by:

$$\forall t \in \tau, \qquad k_i.t_i(t) = \bar{r_i}(t) \leq\ [Capacity\ of\ the\ ramp\ associated\ to\ knob\ i]$$

$$\Rightarrow m_i = \frac{[Capacity\ of\ the\ ramp\ associated\ to\ knob\ i]}{\max_t t_i(t)}$$

We impose therefore:

$$\forall i \in K, \qquad 0 \leq k_i \leq m_i$$

which is equivalent to :

$$\vec{k} \in \mathscr{B} \tag{3}$$

with $\mathscr{B}$ the hyper-cube ("box") defined by :

$$\mathscr{B} = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\kappa \end{bmatrix} \in \mathbb{R}^\kappa \middle| \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leq \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\kappa \end{bmatrix} \leq \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_\kappa \end{bmatrix} \right\}$$

The box $\mathscr{B}$ will be called the *search space* of the knobs.

2) Knob groups and flow balance: We define here objects and notation to describe a simple situation: the knobs are closely monitored by nearby mainline sensors, leading often to a situation where a ramp is the only non-monitored ramp between two mainline sensors. This is equivalent to the ramp being monitored.

We call *segment* the set of links between two consecutive mainline sensors, including the links containing these sensors.
We call *knob group* each set of knobs whose corresponding ramps are connected to the same monitored segment. This definition is illustrated in Fig. 1.
We call *partially monitored segment* the monitored segments associated with a knob group i.e. containing at least one non-monitored ramp.

Denoting γ the total number of knob groups, we have:

Partially monitored segments : $(S_i)_{i\in[\![1,\gamma]\!]}$

Formal definition:
$\exists\,!\,\gamma \in M,\ \exists\,!\,((\beta_i,\eta_i))_{i\in[\![1,\gamma]\!]} \in (T^2)^{[\![1,\gamma]\!]}$ *s.t.*,
denoting $S_i = [\![\beta_i,\eta_i]\!]$ *and* $S = \underset{i\in[\![1,\gamma]\!]}{\cup} S_i$:
$\forall i \in [\![1,\gamma]\!]$,
1) $[\![\beta_i,\eta_i]\!] \cap T = \{\beta_i,\eta_i\}$
2) $[\![\beta_i,\eta_i]\!] \cap K \neq \{\emptyset\}$
and $\forall k \in K,\ k \in S$.

Knob groups : $(g_i)_{i\in[\![1,\gamma]\!]}$

Formal definition:
$\forall i \in [\![1,\gamma]\!],\ g_i = S_i \cap K$

We can deduce the value of the daily flow brought by the knobs of each group from its *knob group flow balance*: the difference between all the flows entering and all the flows exiting their corresponding partially monitored segment. That is the sum of the flow exiting the mainline entrance of the segment and the flows exiting the monitored on-ramps throughout the segment, minus the sum of the flow exiting the mainline exit of the segment and the flows exiting the monitored off-ramps throughout the segment.

Knob group flow balances : $(\Delta_i)_{i\in[\![1,\gamma]\!]}$

Formal definition:

$$\forall i \in [\![1,\gamma]\!], \qquad \Delta_i = \sum_{t\in\tau}\Big[\widetilde{f}_{\beta_i}(t) - \widetilde{f}_{\eta_i}(t) + \sum_{j\in(R\setminus K)\cap S_i} \sigma_j.\widetilde{r_j}(t)\Big]$$

The standard flow balance equation of each partially monitored segment is:

$0 = \sum_{t\in\tau}\Big[f_{\beta_i}(t) - f_{\eta_i}(t) + \sum_{j\in R\cap S_i} \sigma_j.r_j(t)\Big]$ i.e. the sum of all the signed flows entering and exiting the segment is zero.

As stated in eq. 1, $\widetilde{f}_{\beta_i}(t) = f_{\beta_i}(t)$, $\widetilde{f}_{\eta_i}(t) = f_{\eta_i}(t)$ and $\widetilde{r_j}(t) = r_j(t)$. The balance equation becomes therefore:

$$0 = \Delta_i + \sum_{t\in\tau}\Big[\sum_{j\in g_i} \sigma_j.r_j(t)\Big]$$

leading to:

$$\Delta_i = -\sum_{t\in\tau}\Big[\sum_{j\in g_i} \sigma_j.r_j(t)\Big]$$
$$\Leftrightarrow \Delta_i = -\sum_{j\in g_i} \sigma_j\Big[\sum_{t\in\tau} r_j(t)\Big]$$

and thanks to eq. 2:

$$\Leftrightarrow \quad \forall i \in [\![1, \gamma]\!], \quad \Delta_i = -\sum_{j \in g_i} \sigma_j.k_j.\Theta \tag{4}$$

Eq. 4 shows that, for every knob group, the knobs composing it are linked by one linear equation.
This equation determines uniquely the value of the single-knob groups and links the multiple-knob groups with one linear constraint. The next paragraph describes how we apply uncertainties to this equation in order to produce new, closer to reality knob boundaries.

3) Refined knob boundaries: The local uncertainty described in II prevents us from keeping Eq. 4 as a constraint for the parameters.

$\forall i \in [\![1, \gamma]\!]$, let the most permissive boundaries:

$\Delta_i^- = \max\{|\Delta_i| - U^{add}; |\Delta_i|.(1 - U^{mul})\}$
$\Delta_i^+ = \min\{|\Delta_i| + U^{add}; |\Delta_i|.(1 + U^{mul})\}$

Taking the local uncertainty into account in Eq. 4 is translated into the following linear inequality constraints:

$$\forall i \in [\![1, \gamma]\!], \ \Delta_i^- \leq |\sum_{j \in g_i} \sigma_j.k_j.\Theta| \leq \Delta_i^+ \tag{5}$$

These γ inequalities drastically reduce the size of the search space, defining a new *feasible space*.

Comments:
We can now illustrate and justify the form that we have adopted for the uncertainty. This form allows us to quantify the freedom given to the result: the flow balance of each knob-group is between $(1 - U^{mul})$ and $(1 + U^{mul})$ times what has been measured by the mainline sensors, aknowledging that we don't accept less than $\pm U^{add}$ cars precision on the measures.
Taking into account U^{add} is indispensable. This is observed in the case of single-knob groups, where Eq. 4 leads to a unique value for the knob of the group. Let us call it *perfect value of the knob i*, denoted $k_i^{perfect}$. If $U^{add} = 0$, it immediately follows from Eq. 5 that two new boundaries are set for k_i, if they are tighter than $[0, m_i] : k_i \in [(1 \pm U^{mul}).k_i^{perfect}]$.
The problem happens when the perfect value is a ridiculously small quantity. The maximum obtained with $(1 + U^{mul}).k_i^{perfect}$ then corresponds often to a total daily flow of less than 500 cars exiting the ramp, which is not acceptable.

Example: In the scenario we experiment on, one of the ramps has a perfect value of 0.016, which leads to a maximum of 0.032 i.e. 406 cars going through the ramp during the whole day if U^{mul} is set to 100% (very permissive: the daily flow can double what is measured by the mainline sensors). This is way too small for the scenario.
However, in fact, the fluctuation allowed by the new boundaries of this ramp is 406 *cars*, which is 13 times smaller than the sensor uncertainty (i.e. $5\%.[mean\ on\ i \in T\ of\ \widetilde{F_i}] = 5139\ cars$): the sensors do not have this level of precision, and the sensor noise/bias is responsible for this impossible perfect value.
Once $U^{add} = 5\%.[mean\ on\ i \in T\ of\ \widetilde{F_i}]$ is taken into account, the maximum of the knob becomes 0.42, which corresponds to 5329 *cars* and offers an acceptable range to the flow going through the ramp.

C. Performance metrics and error calculators

Three performance metrics are used on the model output to measure the state of the freeway. Each of these metrics is then compared by an error calculator to its value computed on the data. The errors are expressed as a percentage, in order to monitor easily their relative importance. As exposed just below, the common

principle for this percentage computation is to calculate the relative difference between the model output performance and the data performance.

In all that follows, we will abusively denote the functions of the model output as functions of $\vec{k}$, the knobs vector antecedent of the output.

1) Vehicle Hours Travelled (VHT): This quantity is the sum of the time spent on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant.

VHT computation on monitored mainline links output:

$$VHT(\vec{k}) = \frac{dt}{[1\ hour]} \sum_{i \in T} L_i \sum_{t \in \tau} d_i(t)$$

Denoting $\widetilde{VHT}$ the value computed from the data using the same formula, the error is the relative difference :

$$E_{VHT}(\vec{k}) = \frac{|VHT(\vec{k}) - \widetilde{VHT}|}{\widetilde{VHT}}$$

2) Vehicle Miles Travelled (VMT): This quantity is the sum of the distance traveled on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant.

VMT computation on monitored mainline links output:

$$VMT(\vec{k}) = \sum_{i \in T} L_i \sum_{t \in \tau} f_i(t)$$

Denoting $\widetilde{VMT}$ the value computed from the data using the same formula, the error is the relative difference :

$$E_{VMT}(\vec{k}) = \frac{|VMT(\vec{k}) - \widetilde{VMT}|}{\widetilde{VMT}}$$

Reduction of the feasible space: we present here a method used to reduce the feasible space size by forcing the knobs to match the correct VMT value.
VMT is the result of a simple *a priori* calculation that does not need the traffic model output computation, if we know the boundary conditions at $t = 0$ and $t = D$. As exposed in I-B, we assume that these conditions are ≈ 0 $cars$ on every link (D has to be big enough for these conditions to be very small in comparison with the total number of vehicles going through the freeway during D).

Let $VMT^{ref} = VMT(\vec{k}^{ref})$, a certain VMT reference value output by the model. We suppose that $\vec{k}^{ref}$ is some feasible knobs vector (in our case, we used $\vec{k}^{ref} = (1, ..., 1)$).
Denoting $VMT^{apriori}(\vec{k})$ the expected VMT value computed from $\vec{k}$:

$$VMT^{a\ priori}(\vec{k}) = VMT^{ref} + \sum_{i \in K} \left[\sigma_i . k_i . \Theta . \sum_{\substack{j \in T \\ j > i}} L_j \right] \tag{6}$$

The *a-priori* calculation above consists in anticipating the deviation from VMT^{ref} caused by changing the knobs from $\vec{k^{ref}}$ to $\vec{k}$. For each knob, the flow change resulting from its modification is multiplied by the remaining mainline length. These κ contributions are then summed.

The linear equation 6 empowers us to constrain the input $\vec{k}$ in order to ensure $\widetilde{VMT} = VMT^{a\ priori}(\vec{k})(\approx VMT(\vec{k}))$, thus reducing the size of the feasible space by one dimension:

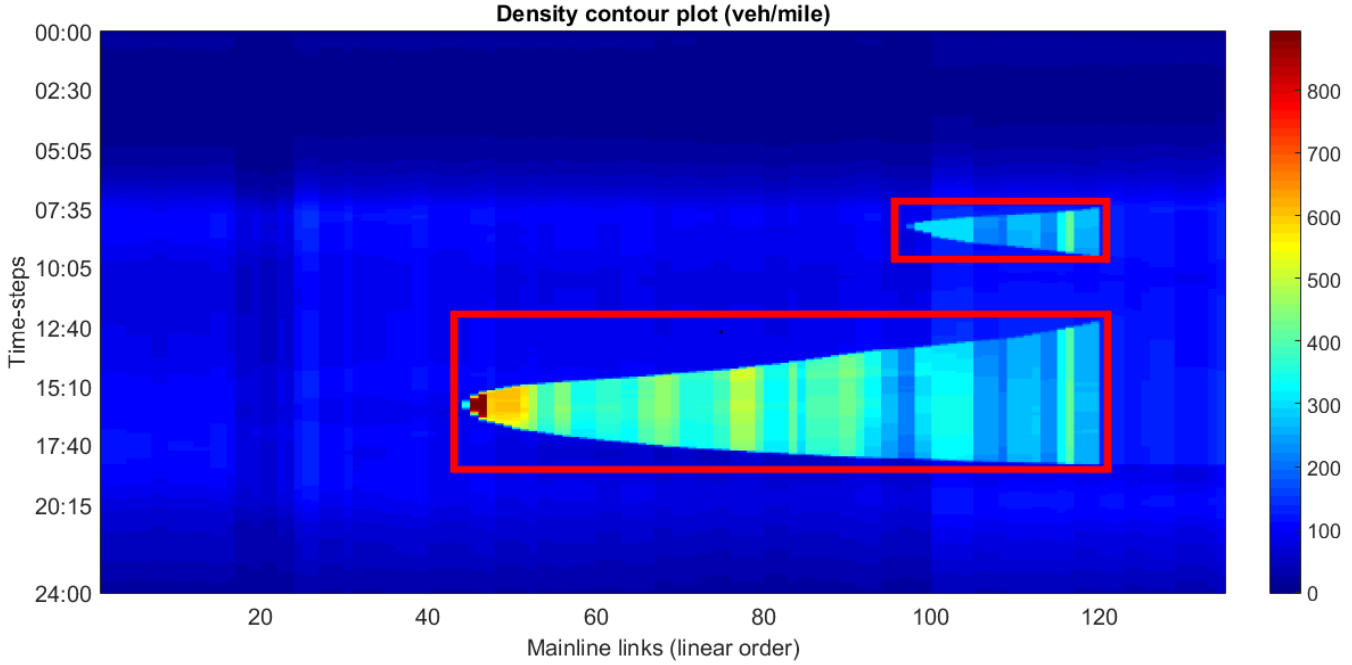
$$\sum_{i\in K}\left[\sigma_i.k_i.\Theta.\sum_{\substack{j\in T\\ j>i}} L_j\right] + VMT^{ref} = \widetilde{VMT}$$

However, the global uncertainty applied on VMT (which is a global quantity computed from the sum over the whole time and space) forces us to loosen this last constraint equation.
Denoting $\widetilde{VMT}^{-} = \widetilde{VMT}.(1 - U^{global})$ and $\widetilde{VMT}^{+} = \widetilde{VMT}.(1 + U^{global})$, it becomes:

$$\widetilde{VMT}^{-} \leq \sum_{i\in K}\left[\sigma_i.k_i.\Theta.\sum_{\substack{j\in T\\ j>i}} L_j\right] + VMT^{ref} \leq \widetilde{VMT}^{+} \tag{7}$$

3) Congestion Pattern: We call *contour plot* the graph representing the value of a quantity on every mainline link at all times: the mainline links as abscissa and time steps as ordinates. Fig. 2. below is an example of a density contour plot for one day on a 135 links freeway, from a traffic model output.

Fig. 2. Example of a density contour plot on a 135 mainline links freeway over 24h out of a traffic model output.



This plot is used to monitor easily where and when the congestion is : here, we see empirically that it is contained in the two framed parts (where the density is much higher).
In what follows, by analogy, we will call *contour domain* the set $\mathscr{P}=\{(i,t)|\ i \in M,\ t \in \tau\}$ and *pixel* each of its elements.
On the model output, we define the *congested* pixels as the ones where the density exceeds some *critical density* deduced for each link from the freeway and traffic models. The main feature of our calibration method is to fit the locations and times of these congested pixels to what the measurements indicate.
For each mainline link $i \in M$, we denote d_i^* the critical density.
We define the *output congested domain* $\mathscr{C} \in \mathscr{P}$ containing the congested pixels :

$$\mathscr{C} = \{(i,t) \in \mathscr{P} \mid d_i(t) \geq d_i^*\}$$

To define an error based on $\mathscr{C}$, a domain supposed to contain the congestion as to be determined. From the *data density contour plot* (partial, obtained only on the monitored links), we define a target domain $\widetilde{\mathscr{C}} \subset \mathscr{P}$

fitting the data congested pixels as best as it can following some criteria (how this domain is built depends on the amount of data the operator possesses and on his goals. In our case, $\widetilde{\mathscr{C}}$ was a rectangle containing all the congestion seen in the data contour plot).

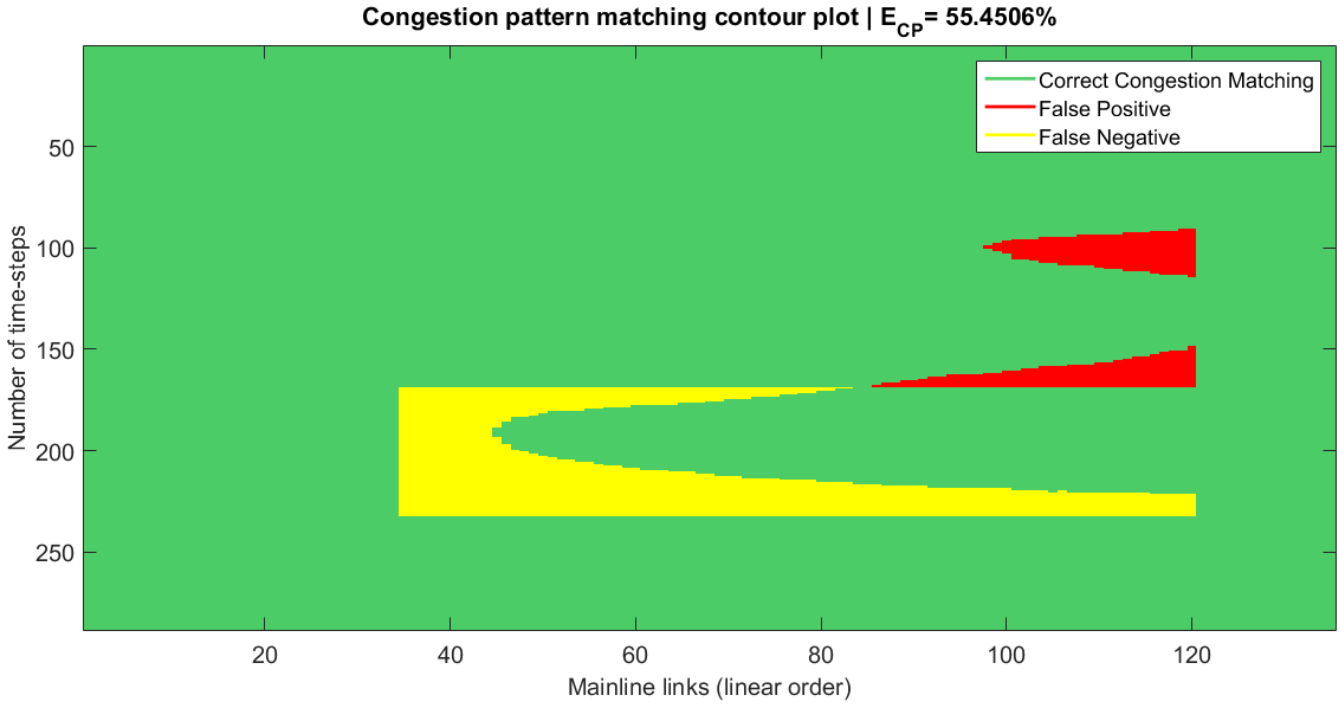
We can now define the congestion pattern error denoted E_{CP} as the normalized number of wrong congestion state pixels. That is, we add one to E_{CP} for each pixel that is not congested but should and for each pixel that is congested but shouldn't. We then divide this result by the number of pixels that should be congested (i.e. $Card(\widetilde{\mathscr{C}}) = \sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{(i,t)\in\widetilde{\mathscr{C}}}$).

$$E_{CP}(\vec{k}) = \frac{\sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{\{(i,t)\in(\widetilde{\mathscr{C}}\setminus\mathscr{C})\cup(\mathscr{C}\setminus\widetilde{\mathscr{C}})\}}}{\sum_{t\in\tau}\sum_{i\in M}\mathbb{1}_{(i,t)\in\widetilde{\mathscr{C}}}}$$

Note that E_{CP} becomes very sensible if the data does not contain much congestion (i.e. $Card(\widetilde{\mathscr{C}}) << Card(\mathscr{P})$).

Fig. 3. below is a visual representation on a contour domain of $(\widetilde{\mathscr{C}}\setminus\mathscr{C})$ otherwise called *false negative pixels*, $(\mathscr{C}\setminus\widetilde{\mathscr{C}})$ otherwise called *false positive pixels* and $(\mathscr{C}\cap\widetilde{\mathscr{C}})\cup(\mathscr{P}\setminus(\mathscr{C}\cup\widetilde{\mathscr{C}}))$, which are the *correct congestion matching* pixels.
Here, $\widetilde{\mathscr{C}}$ is the yellow rectangle.

Fig. 3. Visual example of the congestion pattern matching computation. $\widetilde{\mathscr{C}}$ is one rectangle.



D. Objective function

The calibration method consists in minimizing jointly the three errors described in III-C. We accomplish this goal by minimizing an *objective function* Φ, which is the weighted sum of the errors modified by the global uncertainty, as explained below.

Uncertainty handling : U^{global}, described in I-C, defines a tolerance threshold for the error results. The error results below U^{global} are set to zero, in order to avoid any discrimination between them (we do not have a level of precision below U^{global}).

Each of the three errors E will therefore be multiplied by $\mathbb{1}_{(E>U^{global})}$.

Let $(w_i)_{i\in[\![1,3]\!]}$ the weights, verifying :
$w_1 + w_2 + w_3 = 1$ *and* $\forall i \in \{1,2,3\},\ w_i \geq 0$.

Let the three error *contributions*:
$\phi_{VHT}(\vec{k}) = w_1.E_{VHT}(\vec{k}).\mathbb{1}_{(E_{VHT}>U^{global})}$
$\phi_{VMT}(\vec{k}) = w_2.E_{VMT}(\vec{k}).\mathbb{1}_{(E_{VMT}>U^{global})}$
$\phi_{CP}(\vec{k}) = w_3.E_{CP}(\vec{k}).\mathbb{1}_{(E_{CP}>U^{global})}$

Φ is defined by:

$$\Phi : \left| \begin{array}{ccl} \mathscr{B} & \longrightarrow & [0,100] \\ \vec{k} & \longmapsto & \phi_{VHT}(\vec{k}) + \phi_{VMT}(\vec{k}) + \phi_{CP}(\vec{k}) \end{array} \right.$$

Note : the errors can lead to values superior to 1 thus giving values of Φ *superior to 100% but, to simplify, we will ignore these cases that are very far from the objective.*

This definition as a weighted sum of percentages implies that the values of Φ can be interpreted as a *global error percentage*. Thanks to the normalization of the errors, the weight given to each component is equivalent to the importance the operator wants to give to each one of them.

E. Optimization problem statement

The optimization problem that we solve can now be stated :

$$\begin{aligned} & minimize \quad \Phi(\vec{k}) \\ & s.t. \quad \forall i \in [\![1,\gamma]\!],\ \Delta_i^- \leq |\sum_{j\in g_i} \sigma_j.k_j.\Theta| \leq \Delta_i^+ \\ & and\ \widetilde{VMT}^- \leq \sum_{i\in K}\left[\sigma_i.k_i.\Theta.\sum_{\substack{j\in T \\ j>i}} L_j\right] + VMT^{ref} \leq \widetilde{VMT}^+ \\ & and\ \vec{k} \in \mathscr{B} \end{aligned} \tag{8}$$

IV. Numerical method

A. Requirements

The performance errors are irregular functions. In particular, the congestion pattern fitting reflects congestion phenomena. These present numerous thresholds in their non-smooth behavior.
We can also point out that these errors aren't always correlated.
Furthermore, each evaluation of the error function requires the execution of a simulation (around 5 seconds on a desktop computer), and this evaluation is the only thing accessible of Φ: there is no way of quickly computing its value or its gradient .
Therefore, the objective function is a black box.
We deduce from these observations that convex optimization methods and derivative-based methods are not adapted to our case.
The search space is a continuous hyper-cube, as explained in III-B1.
We can conclude that we study a non-linear, non-convex black-box imputation problem in continuous domain.

In addition, the resolution method has to be adaptive, since it will be applied to many different freeways,

times and sensor densities. We want as few numerical method parameters to tune as possible and no prior optimization knowledge required if possible.

Finally, our approach does not take execution time as a criteria : the goal is to obtain the best possible result quality and uniqueness (global minimum of Φ).

B. Co-variance Matrix Adaptation - Evolution Strategy (CMA-ES)

The state-of-the-art evolutionnary algorithm CMA-ES is very well suited for these requirements.
Its architect, Nikolaus Hansen (see [1]), describes it in these words :

- It is conceived to solve "difficult non-linear non-convex black-box optimisation problems in continuous domain".
- It is feasible on "non-smooth and even non-continuous problems, as well as on multimodal and/or noisy problems".
- "The CMA-ES does not use or approximate gradients and does not even presume or require their existence"
- It is "competitive for global optimization".
- It is adapted to search spaces of dimension between 3 and 100.

In addition, it is extremely adaptive as only an initial standard deviation σ and the population size λ have to be tuned.
A more precise description of CMA-ES can be found in [2] and [3], especially in the parts *0.3: Randomized Black-Box optimization* and *5: Discussion*.
For further understanding, the reader can keep in memory that the algorithm samples a population Π_p of λ random points at iteration p. It then evaluates each one of them, and modifies its internal parameters so that the next λ sampled points Π_{p+1} will be sampled more probably in the direction of the points of Π_p that gave the smaller Φ values. It globally keeps memory of the fitness (objective function value) of the points it encountered.
In what follows, the objective function Φ will also be called *fitness function*.

It is recommended to give the same sensitivity to the parameters i.e., in our case, to give the same range to the knobs (we rescaled each one of them to a 0-10 range before imputation to CMA-ES).
In addition, it is recommended to set σ in the range $[0.2, 0.5]$ times the size of the knobs range ($[2, 5]$ in our case).
Finally, the recommended first value to give to λ is $4 + \lfloor 3.log(\kappa) \rfloor$.

C. Constraints implementation

The CMA-ES source code handles box constraints natively using the "repair and penalize" procedure. This consists in repairing non-feasible sampled points before inputting them to the model; and penalizing (i.e. increasing the value of the fitness function) proportionally to the distance between the unfeasible point and the feasible space. This method forces the algorithm to eventually enter and stay in the feasible domain while avoiding evaluating non-allowed points.
However, the source code does not handle linear constraints. We describe here how we apply manually this repair & penalize procedure to reflect the two linear constraints shown in III-E.

Repairing:

Let $\vec{\underline{k}}^{(p)}$ the knobs vector sampled by CMA-ES at iteration p (i.e. before repairment).
The projection is implemented using the following program in a standard quadratic optimization solver:

$$
\begin{aligned}
&minimize \quad \left\| \vec{k}^{(p)} - \vec{\underline{k}}^{(p)} \right\|_2 \\
&s.t. \quad \forall i \in G, \;\; \Delta_i^- < |\textstyle\sum_{j \in g_i} \sigma_j.k_j.\Theta| < \Delta_i^+ \\
&and \; \widetilde{VMT}^- \leq \sum_{i \in K} \left[\sigma_i.k_i.\Theta. \sum_{\substack{j \in T \\ j>i}} L_j \right] + VMT^{ref} \leq \widetilde{VMT}^+ \\
&and \quad \vec{k}^{(p)} \in \mathscr{B}
\end{aligned}
$$

Let us illustrate the effect of this program with an example. We suppose that the $\widetilde{VMT}^{\pm}$ condition above is enough loose for it to be respected without influencing the projection (often verified in practice). Fig. 4 below

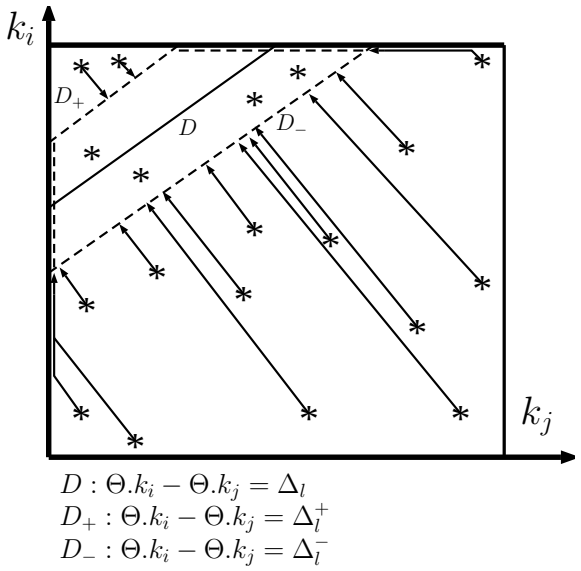


Fig. 4. Example of repair projections on a two-knobs group.

displays, for a two-knobs group $g_l = \{i, j\}$, the projection of several sampled points due to the two remaining conditions ($\Delta_l^{\pm}$ and $\mathscr{B}$). D is the hyper-plan (straight line in dimension 2) on which the knob group flow will have the exact value Δ_l. The two *tolerance hyper-plans* on which the knob group flow value will be Δ_l^+ and Δ_l^- are respectively D^+ and D^-. In addition, the external square is the hyper-cube corresponding to the physical boundaries of the two knobs (from $\mathscr{B}$). The points sampled in the feasible space (the doted line trapezium) remain untouched while the others are projected on the nearest point of the edge of the feasible space.
This repair ensures that only feasible values of the input are tested and that the algorithm does not get stuck in an unfeasible hole of the physical boundaries hyper-cube.

Penalizing:

At each evaluation, a penalization proportional to the distance between the projected and original point is added to the fitness function. This ensures that the algorithm will come closer to the feasible space at every iteration until eventually entering and staying inside it. This feature is important as it avoids two imbalances on the sampling:

- Testing more points on the edges than we should: if the algorithm is left sampling points far from the edges, without penalization, it will have no incentive to prefer sampling next to the feasible space than far. This is an obstacle from entering the feasible space, as all the points on the straight lines perpendicular

to the edges would be equivalent. This would lead to a situation where it is common that too much (or all) of the points are sampled on the edges while CMA-ES converges far from the feasible space.

- Imbalance between the edge points tested: the edge points which are the projection of more unfeasible hyper-cube points than others will be sampled unfairly more often.
  Example: in Fig. 4, the points of the square on the mediator of the segment formed by the intersections of D and the square are more numerous than the points on any other straight with same slope.
  The points on one of these straights and below the feasible space are all projected to the same point of the edge of the feasible space. Therefore, without penalization, the middle of the segment cited above would be sampled more often than the other points of its edge, for a reason that is not the simulation output it leads to.

The penalization, denoted E_{proj}, is normalized by a factor which is the distance between the physical maximums and minimums vectors, reflecting the *order of magnitude* of the search space.

$$E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)}) = \frac{\left\| \vec{k}^{(p)} - \underline{\vec{k}}^{(p)} \right\|_2}{\left\| \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_\kappa \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right\|_2}$$

A fourth contribution $\phi_{proj} = w_4.E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)})$ is therefore added to the fitness function.

With the same definitions as in III-D and
$\sum_{i=1}^{4} w_i = 1$ *and* $\forall i \in [\![1,4]\!]\ w_i \geq 0$,

the final fitness function used is J defined by :

$$J : \left| \begin{array}{ccc} \mathscr{B} & \longrightarrow & [0,100] \\ \underline{\vec{k}}^{(p)} & \longmapsto & \Phi(\vec{k}^{(p)}) + w_4.E_{proj}(\underline{\vec{k}}^{(p)}, \vec{k}^{(p)}) \end{array} \right.$$

We denote J^* the minimum of J encountered by CMA-ES during its search :

$$J^* = \min_p J(\vec{k}^{(p)})$$

Similarly, all values (input or output) denoted with stars are the ones corresponding to the J evaluation that gave J^*.

Single-knob group specificity: For a single-knob group, the condition Eq. 5 is equivalent to hard boundaries for the concerned knob (one equation in one dimension). In this case, the projection due to Eq. 5 is not implemented but the physical boundaries (from $\mathscr{B}$) of the knob input to CMA-ES are replaced by these new boundaries, if they are narrower :

$\forall\ i \in G\ s.t.\ Card(g_i) = 1,\ i.e.\ g_i = \{j\}\ :$

$$\frac{\max\{0; |\Delta_i^-|\}}{\Theta} \leq k_j \leq \frac{\min\{m_j; |\Delta_i^+|\}}{\Theta} \tag{9}$$

V. Experiment settings

We present here the elements we chose to run the experiment we made.

A. Context: Origin of the data

The real scenario used for this study is a portion of freeway 210 East in the suburbs of Los Angeles. The measurements used are collected by the sensors network PeMS of Caltrans. For more information, see http://pems.dot.ca.gov/
These daily measurements are flow, density and speed on 74 links of 188, every 5 minutes for 24 hours (289 time-steps).
After deleting partial or too biased data, the sensors have the following distribution:

- 33/135 monitored mainline links
- 26/28 monitored on-ramps
- 15/25 monitored off-ramps

This makes a total of **12 knobs**, that are distributed all along the freeway portion.

We chose to calibrate the model on the average of 5 Tuesdays (0am-12pm) in fall 2014 data. The goal is to find the set of knobs that best fits the profile of a Tuesday (the general shape of the traffic depending greatly on the day of the week).

For each knob-ramp, the template is built by taking the average of the flow profiles of the two closest monitored ramps that have the same incoming traffic context.

B. The traffic model : CTM

We use the Cell Transmission Model, a popular model for macroscopic traffic prediction by Carlos Daganzo (see. [4]).
It consists in solving the kinematic wave equation.
CTM defines the notion of *Fundamental Diagram* for every link, that contains all the traffic properties of the link in 3 parameters : capacity, congestion speed and free-flow speed. The theoretical density congestion threshold of each link is defined by $\frac{Link\ capacity}{Link\ freeflow\ speed}$.

C. The large traffic simulator : BeATS

The macroscopic traffic simulator used is the Berkeley Advanced Traffic Simulator (BeATS). It computes the results of CTM in a mode where the entry, on-ramp AND off-ramp demands are inputs (the *split-ratios* are therefore outputs). The inputs of the simulator are:

- Freeway model.
- Fundamental Diagram of every link.
- Exit flow demand of every source and off-ramp. Time-step is 5 minutes, as for the data.

The outputs are the entry and exit flows, density and average speed in every link, every 5 minutes, for the whole day.

D. Choice of the weights

CMA-ES is very efficient : the choice of the weights does not seem to be very important, if it is not too unbalanced.
Because VMT is always correct by input pre-processing (see III-C2), we attribute it the weight zero.
To insist on the congestion pattern matching, the main feature of our method, we have chosen the following weights:

$$w_1 = 0.5,\ w_2 = 0.25,\ w_3 = 0,\ w_4 = 0.25$$

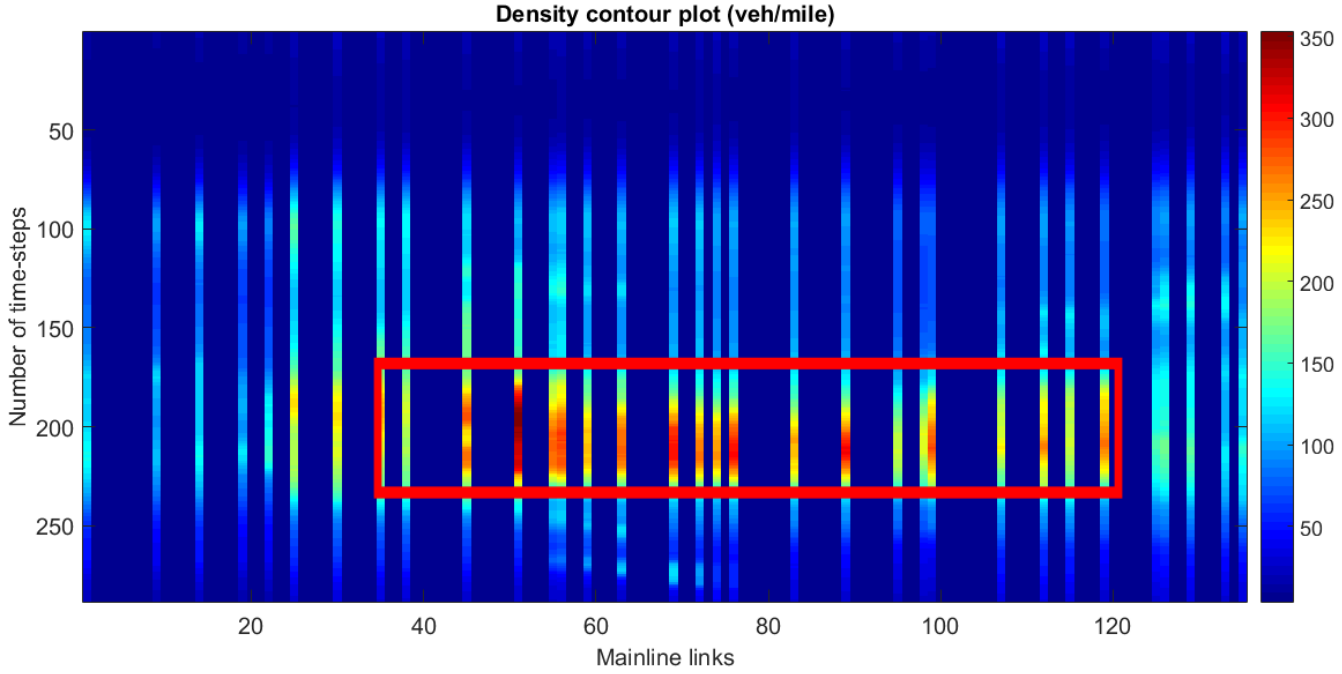
E. Implementation of the congestion pattern

The theoretical congestion threshold d_i^* is not satisfactory as it monitors "noisy" congestion (situations of free-flow that is at the limit of the congestion), distorting E_{CP} results. We add a small number of vehicles δ (in our case, $\delta = 6$) to this threshold in order to monitor only the hardcore congestion that our congested domain $\widetilde{\mathscr{C}}$ is supposed to capture.

$$d_i^* = \frac{Link\ capacity}{Link\ freeflow\ speed} + \delta$$

As we are trying to fit an average congestion, we define the data congested domain $\widetilde{\mathscr{C}}$ as a set of boxes that roughly capture each congestion structure.
The average data density contour plot and the unique box we defined in our case are displayed in Fig. 5 :

Fig. 5. Density contour plot on the average data from 5 Tuesdays. Dark blue vertical strips are un-monitored mainline links.



VI. Experiment results

A. General observations:

With $\kappa = 12$ knobs and the standard CMA-ES population size $\lambda = 11 = 4 + \lfloor 3.log(\kappa) \rfloor$, the algorithm usually lasts less than 2000 *BeATS evaluations* (i.e. 180 *generations*) to converge, which is approximately 3 hours on a regular computer without parallelization. U^{global} is set to a reasonable 5%, in order to give some space to CMA-ES on the global quantities.

Denoting $\widetilde{F}$ the average over $i \in T$ of $\widetilde{F_i}$, we define U^{add}, the mainline sensor confidence interval uncertainty, as a percentage of $\widetilde{F}$. For example, $\frac{U^{add}}{\widetilde{F}} = 5\% \Rightarrow U^{add} = 5139$ *cars*. The main observation, that we will develop later, is that the templates shape limit the quality of the result : in every experiment, not one but two congestion profiles appear on the contour domain and are correlated (i.e. when one grows, the other one too : they are caused by the same knobs). $\mathscr{C}$ matching the unique box $\widetilde{\mathscr{C}}$ implies that the undesirable congestion profile is significant and outside the box.

We will judge the quality of the results of an experiment by the value of J^* *and* the likelyhood of the shape of the congested domain $\mathscr{C}$.

It is important to highlight that, for a given set of parameters, even tough the result quality is equivalent from one algorithm execution to another, the solution input k^* found are very different. The solution of our calibration process is far from being unique for this experiment.

We will therefore say abusively that there are several "equivalent global minima" for J.

Changing the parameters:

Fig. 6. below contains the parameters and global error results of the experiments we made. We monitored the effect of each parameter by changing it slightly from one execution to another. Going in the details of each execution, we will expose in the following sub-sections the effect of these parameters on the result quality. The exact values of the parameters in our case are not relevant as they depend on the scenario, time, sensor distribution, and data quality : this study is only qualitative.

Fig. 6. Experiments parameters and J final value.

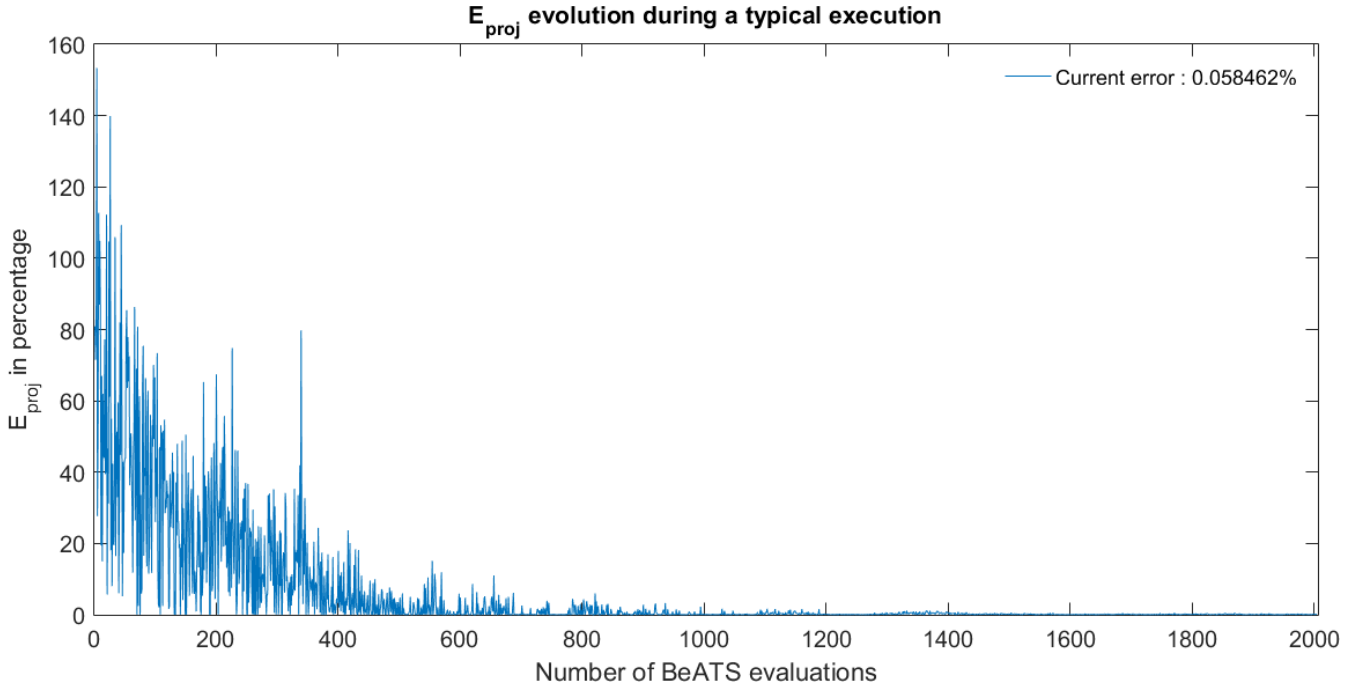
U^{global}	Population Size : λ	Initial standard deviation : σ	$\frac{U^{add}}{\widetilde{F}}$	U^{mul}	Number of BeATS Evaluations	Number of CMA-ES generations	J minimum : J*
5%	11	2	2.5%	25%	2004	182	**43.5%**
				50%			**28.2%**
				75%			**25.4%**
				100%			**23.7%**
			5.0%	25%			**35.1%**
				50%			**25.3%**
				75%			**25.3%**
				100%			**22.8%**
			10.0%	25%			**22.2%**
				50%			**21.6%**
				75%			**21.9%**
				100%			**21.8%**
		5	2.5%	25%			**44.7%**
				50%			**27.3%**
				75%			**27.0%**
				100%			**23.5%**
			5.0%	25%			**36.1%**
				50%			**24.5%**
				75%			**23.7%**
				100%			**25.5%**
			10.0%	25%			**22.8%**
				50%			**22.2%**
				75%			**23.8%**
				100%			**21.4%**
	12	2	2.5%	50%	3002	250	**27.1%**
							27.8%
							27.7%
	24				3002	125	**27.0%**
							27.1%
							27.3%
	36				3026	84	**27.1%**
							27.1%
							27.2%

B. Typical execution

If the parameters are in an wide acceptable range that we will define later, all the execution are alike, because the limiting factor is the shape of the templates.This prevents the algorithm to surpass a certain level of quality, and makes many different parameters combinations to be equivalent in terms of result quality.
We describe here the behavior of the errors during one of these typical equivalent runs (parameters : $\sigma = 5$, $\lambda = 11$, $U^{global} = 5\%$, $\frac{U^{add}}{\widetilde{F}} = 5\%$, $U^{mul} = 75\%$).

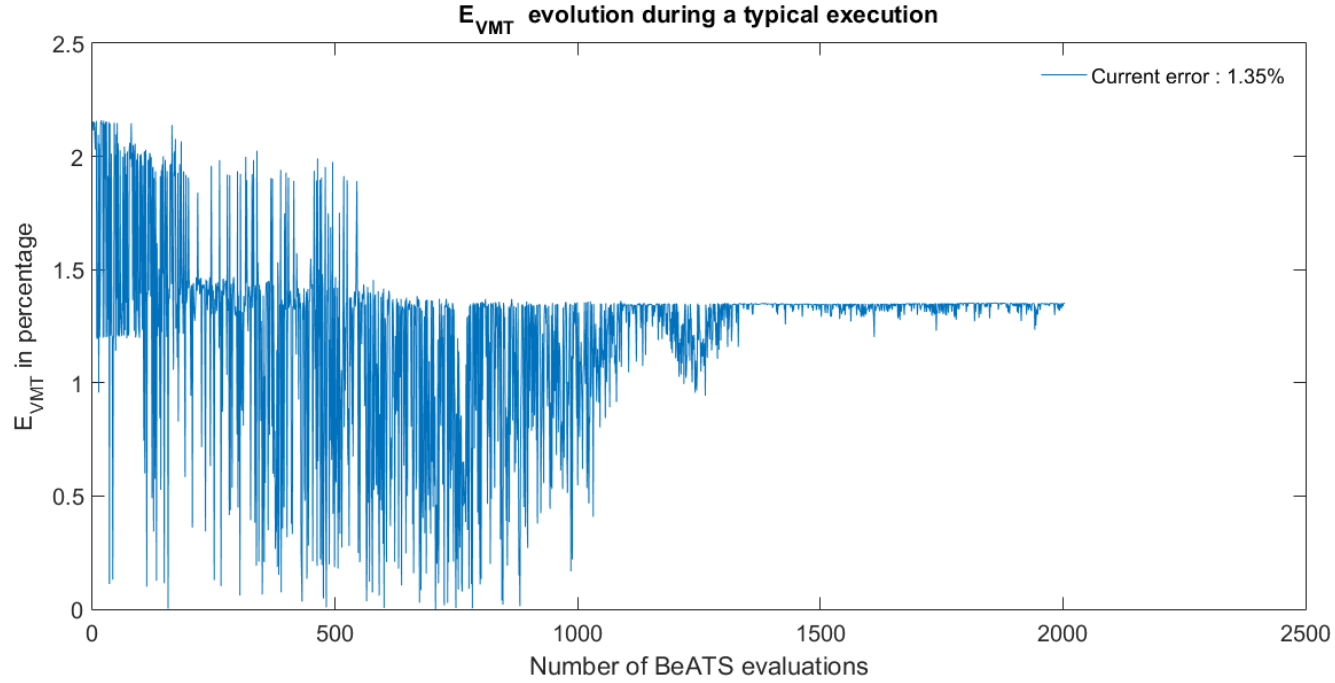
E_{proj}: By construction, E_{proj} is always very close to zero at the end of the execution (the algorithm come closer and closer to the feasible space).
Fig. 7. illustrates this behavior on a typical example.

Fig. 7. Evolution of E_{proj} during a typical execution



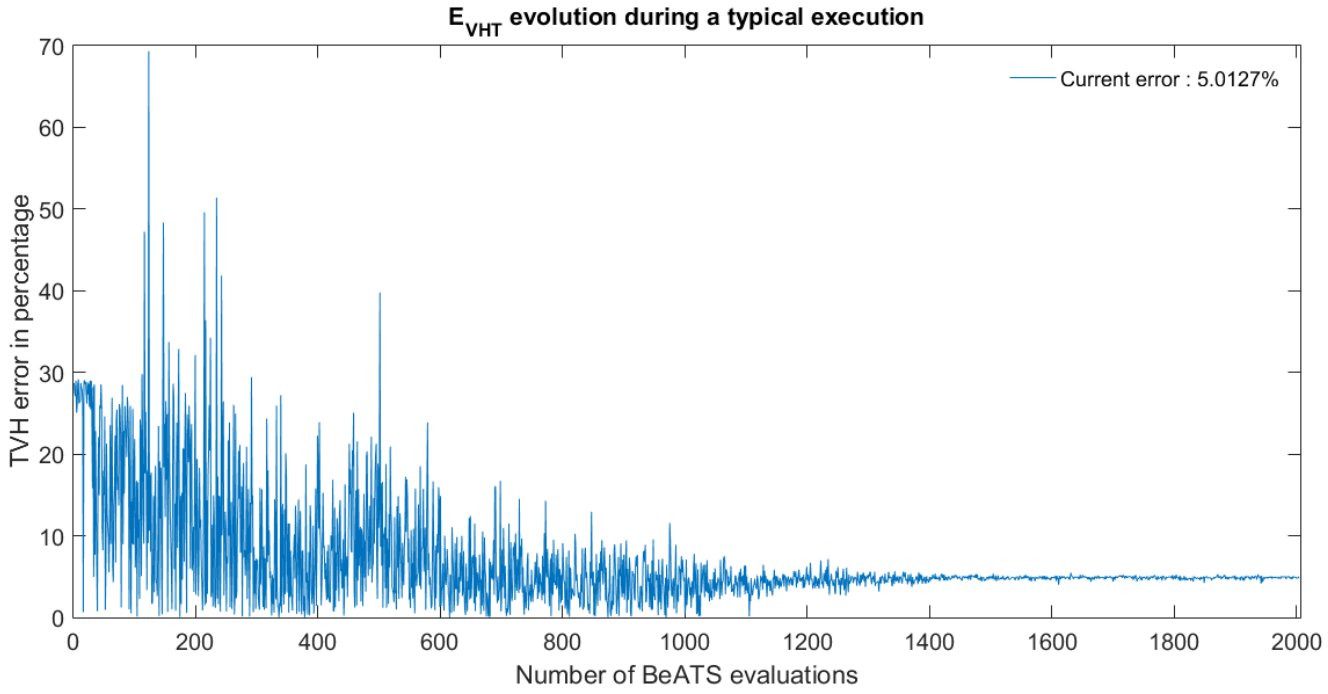
E_{VMT}: By imputation, as explained in III-C2 and IV-C, E_{VMT} is always below 5%, often around $1-2\%$ on each iteration of the algorithm.
Fig. 8 reflects the history of VMT during a typical run.

Fig. 8. Evolution of E_{VMT} during a typical execution



E_{VHT}: VHT is quite correlated with the congestion : on the typical executions that lead to an acceptable congestion, E^*_{VHT} is around $U^{global} = 5\%$. However, particularly when the knobs are constrained too much, E^*_{VHT} ends up around 20%. Fig. 9 reflects the history of E_{VHT} with its typical -good- behavior.

Fig. 9. Evolution of VHT during a typical execution



E_{CP}: E_{CP} behaves like E_{VHT}. The congestion shape obtained for the typical run (best we got) is displayed in Fig. 10. This shape is obtained for all the executions that have acceptable parameters. We can see the two congestion patterns induced by the templates : the smaller one in the morning is correlated with the bigger one in the afternoon and impossible to suppress.

Fig. 10. Typical congestion pattern fitting result

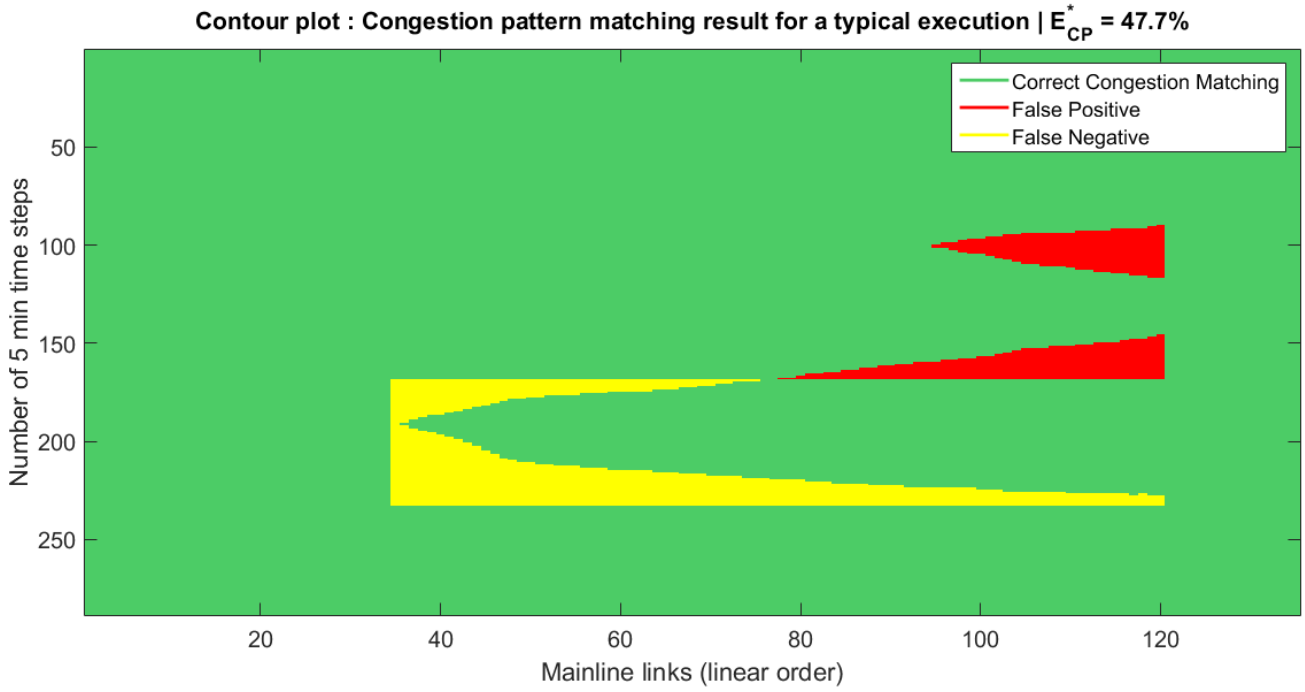


Fig. 11 displays the evolution of the contributions ϕ_{CP}, ϕ_{VMT}, ϕ_{VHT} and ϕ_{proj} of every parameter and their sum, the global error Φ.

Fig. 11. Φ and contributions evolution during a typical execution

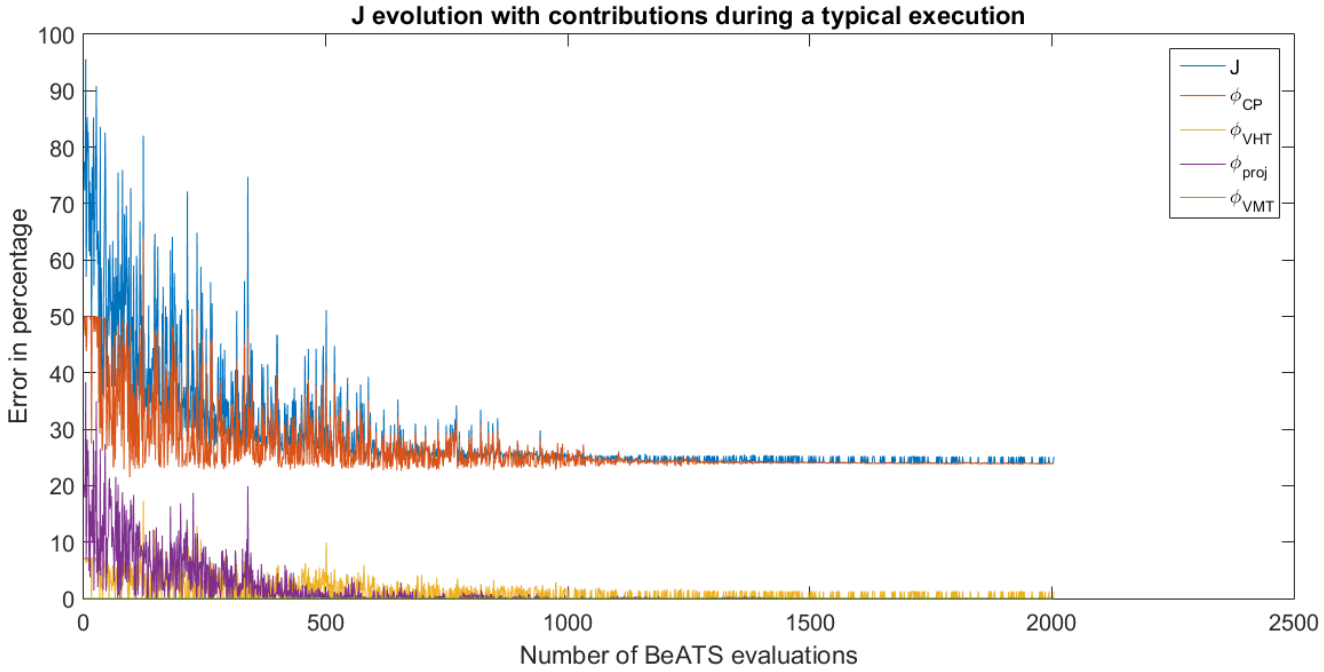
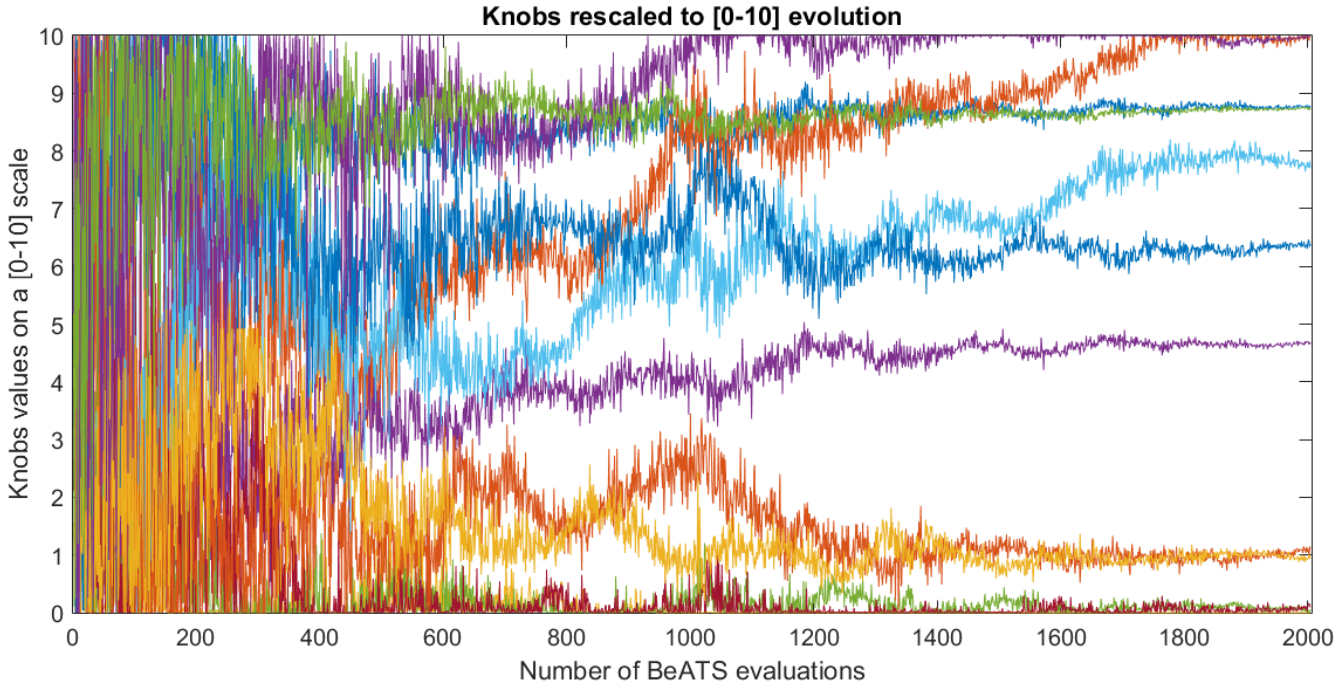


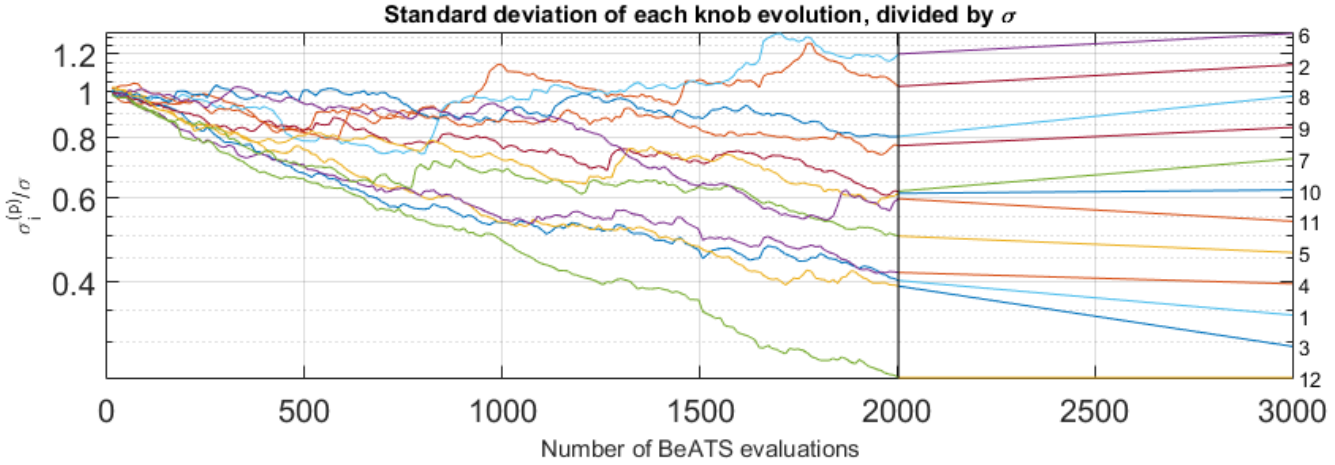
Fig. 12 shows the evolution of the knobs during the execution of CMA-ES (we will call it "knobs history").

Fig. 12. [0-10] knobs history during a typical CMA-ES execution.



Finally, the following figure 13 displays the evolution of the standard deviations associated in CMA-ES to each knob.

Fig. 13. CMA-ES parameters evolution during a typical execution

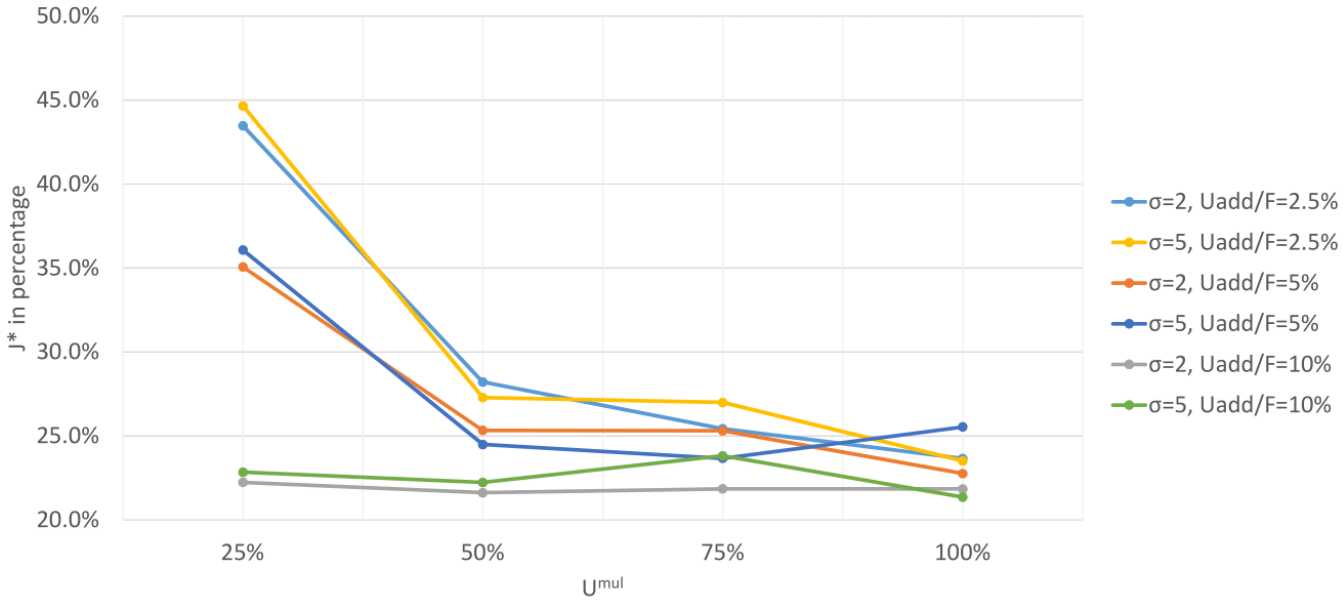


C. Changing U^{mul}

For the following experiments, $\lambda = 11$.

Fig. 14 below shows the evolution of the value of J^* with U_{mul}.

Fig. 14. Evolution of J^* with U^{mul}



As intuitively guessed, the more uncertainty i.e. freedom to the knobs there is, the smaller J^* is.
More precisely, the tendencies observed are the following :

- If U^{add} is not too big (i.e. doesn't override the effect of U^{mul}), there is a minimum threshold on U^{mul} for J^* to be reasonably small : in our case, U^{mul} around 25% doesn't give satisfactory J^* values (35% and 45%). It enters in an acceptable range if $U^{mul} \geq 50\%$. This shows the importance of the uncertainty, due to the lack of precision of the whole : a multiplicative uncertainty of 25% or less is not enough.
- There is no real difference for the effect of U^{mul} between 50% and 75%: they are the typical executions.

- When the multiplicative uncertainty is very big, i.e. $U^{mul} = 100\%$ (minimums are zero i.e. the physical minimums; and the maximums correspond to twice the partially monitored segments flow demands), the result seem to slightly improve. However, one run ($\sigma = 5$ and $\frac{U^{add}}{\widetilde{F}} = 5\%$) shows that it is not always the case : the feasible space becomes very big and CMA-ES is not guided at all. In fact, as we will observe on the congestion pattern contour plots in this case, the results are not satisfactory.

Below are 4 groups of figures. They display especially, for each of the 4 values of U^{mul}, how the resulting congestion pattern fitting and the history of the knobs trough the execution evolve with U^{add} and σ.

Fig. 15. $\mathbf{U_{mul}} = \mathbf{25}\%$. Congestion pattern matching plots : evolution with U_{add} and σ.

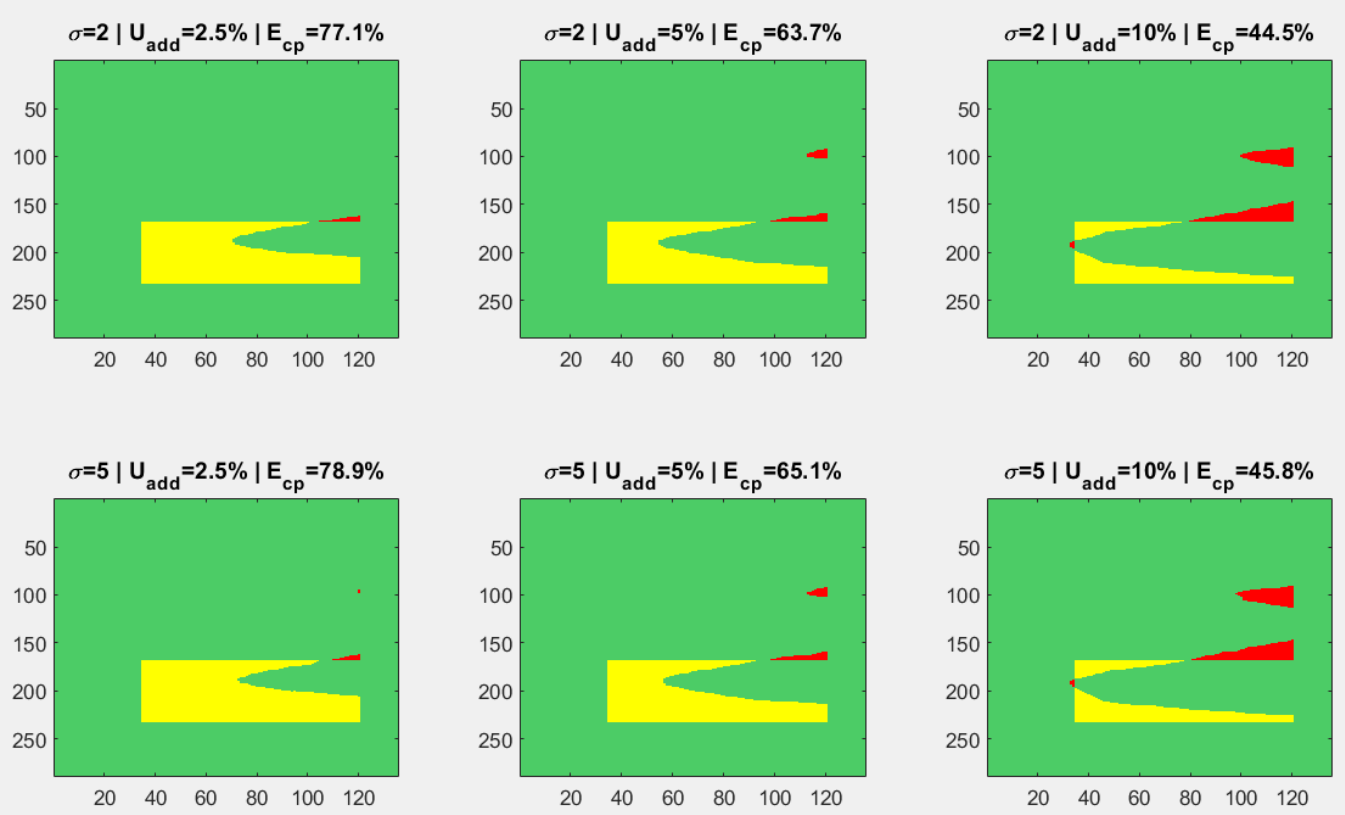


Fig. 16. $\mathbf{U_{mul}} = \mathbf{25}\%$. Knobs history plots : evolution with U_{add} and σ.

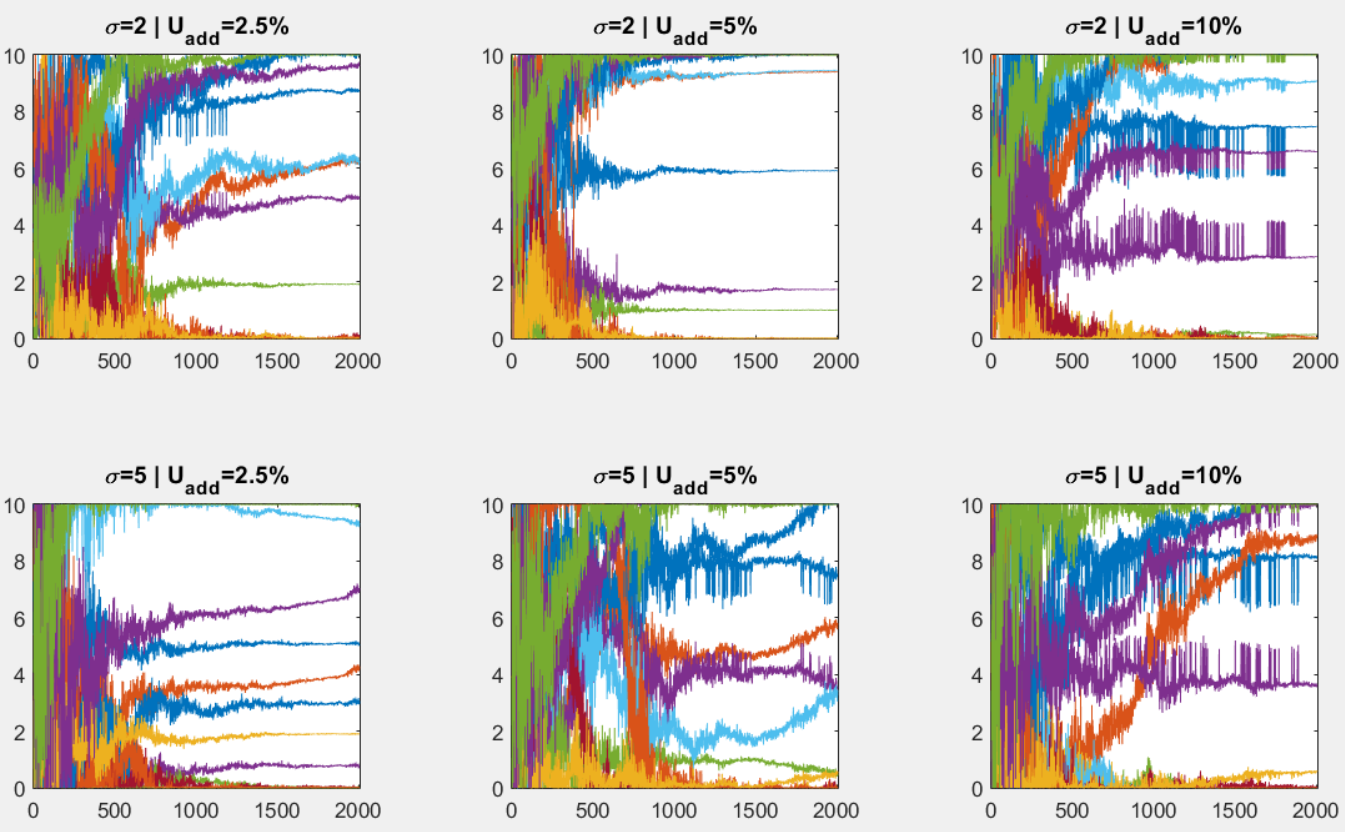
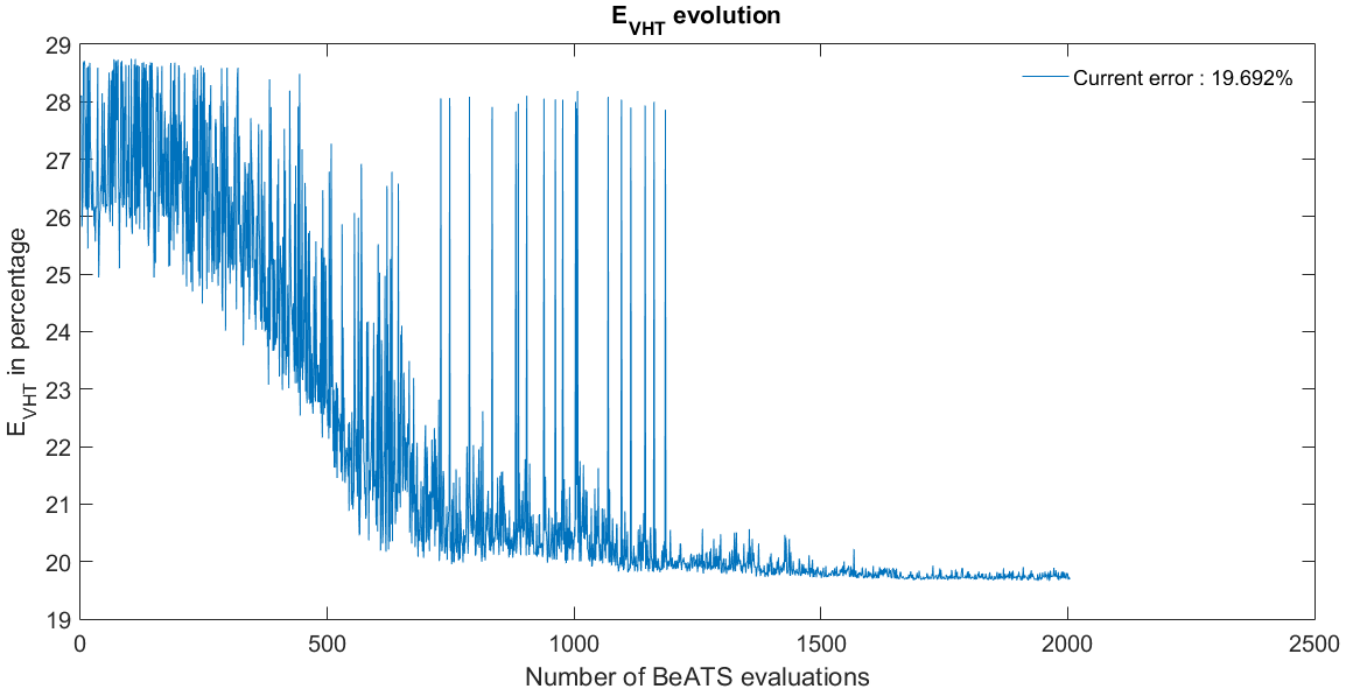


Fig. 17. $\mathbf{U_{mul}} = \mathbf{25}\%$. E_{VHT} evolution plot with $U^{add} = 2.5\%$.



$U^{mul} = 25\%$: Fig. 15, with $U^{mul} = 25\%$, shows the effect on the result of excessively tight constraints. With $U^{add} = 2.5\%$ and 5% (two first columns), $\mathscr{C}$ cannot grow enough to fill $\widetilde{\mathscr{C}}$, which is responsible for a very high E^*_{CP}.

In addition, Fig. 17 shows that, with these parameters, E_{VHT} converges to an unacceptable 20% (it converges below 5% in all other configurations). In the last column, $U^{add} = 10\%$ gives a good space to all the knobs (many of them having their physical boundaries $\mathscr{B}$). In this case, $U^{mul} = 25\%$ has no effect, which is the reason for the good "typical" result obtained.
To summarize, as highlighted previously by Fig. 14, setting U^{mul} to a value smaller than a significant threshold (25% is quite high !) expels the good acceptable solutions from the feasible space.

There is nothing clear to conclude from the knobs evolution on Fig. 16., except that many end up on their boundaries (up to 8 of 12 !), which reflects that they were set too tight i.e. that we describe a far-from-reality situation.

Fig. 18. $\mathbf{U_{mul}} = \mathbf{50}\%$. Congestion pattern matching plots : evolution with U_{add} and σ.

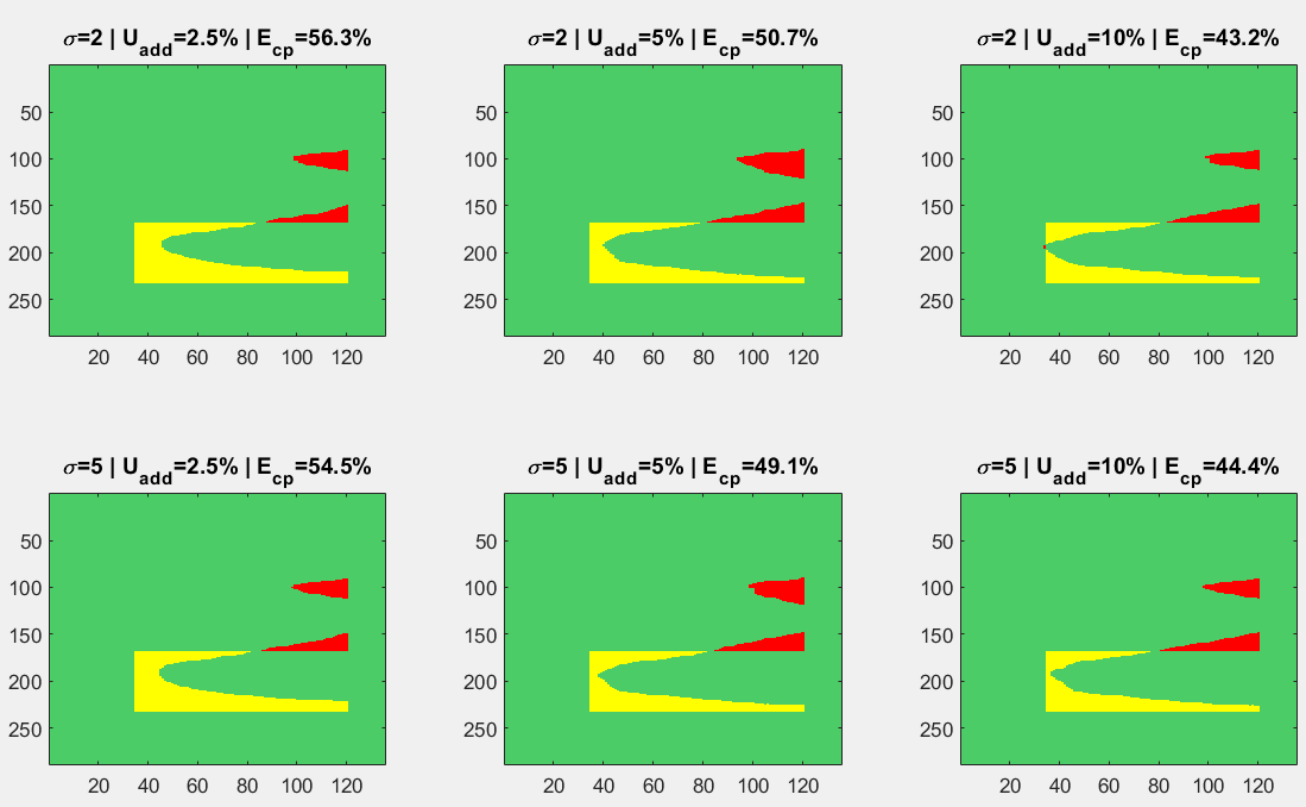
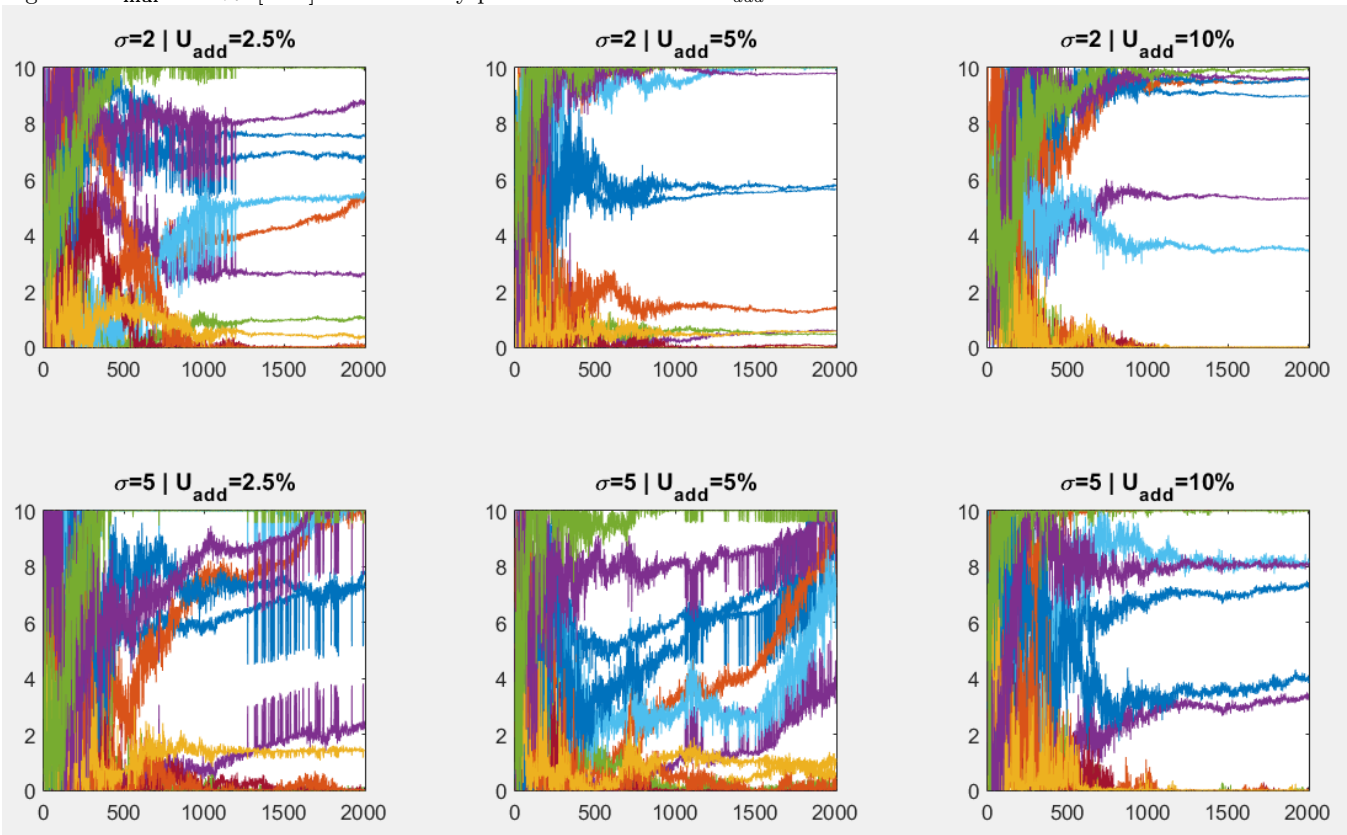


Fig. 19. $\mathbf{U_{mul}} = \mathbf{50}\%$. [0-10] knobs history plots : evolution with U_{add} and σ.



$U^{mul} = 50\%$: We observe in Fig. 18 only typical results, with, as expected, a slight improvement while loosing the U^{add} constraint.
In Fig. 19, there are more situations than previously with many knobs converging to another value than their boundaries. We see a situation (middle-down figure) where the knobs did not finish converging, although the result is good : this illustrates the "equivalent global minima" mentioned in VI-A. Some knobs whose action do not affect $\mathscr{C}$ can evolve in a correlated way (e.g. one on-ramp and one off-ramp knob both increase their value), maintaining all the contributions on a constant value. Their variations are therefore invisible to $\mathscr{C}$ and J.

Fig. 20. $\mathbf{U^{mul}} = \mathbf{75}\%$. Congestion pattern matching plots : evolution with U^{add} and σ.

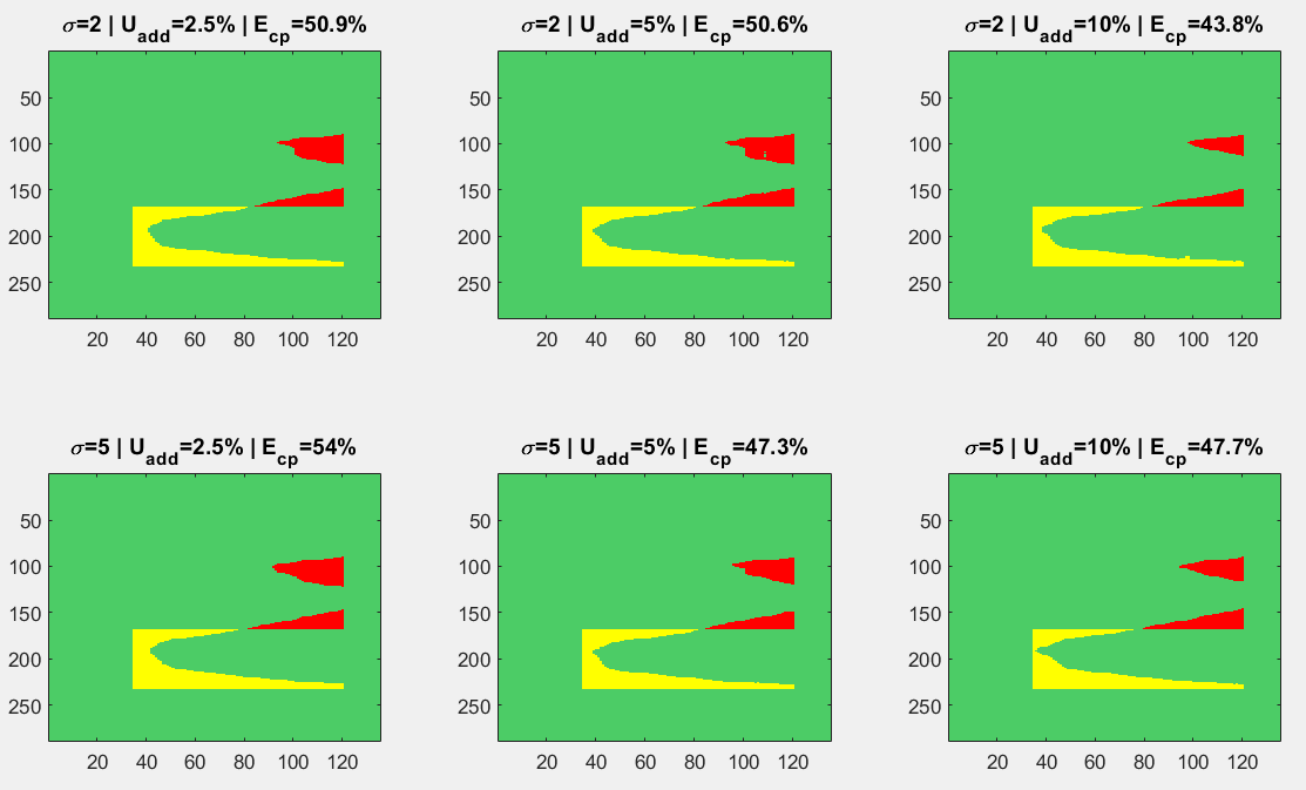
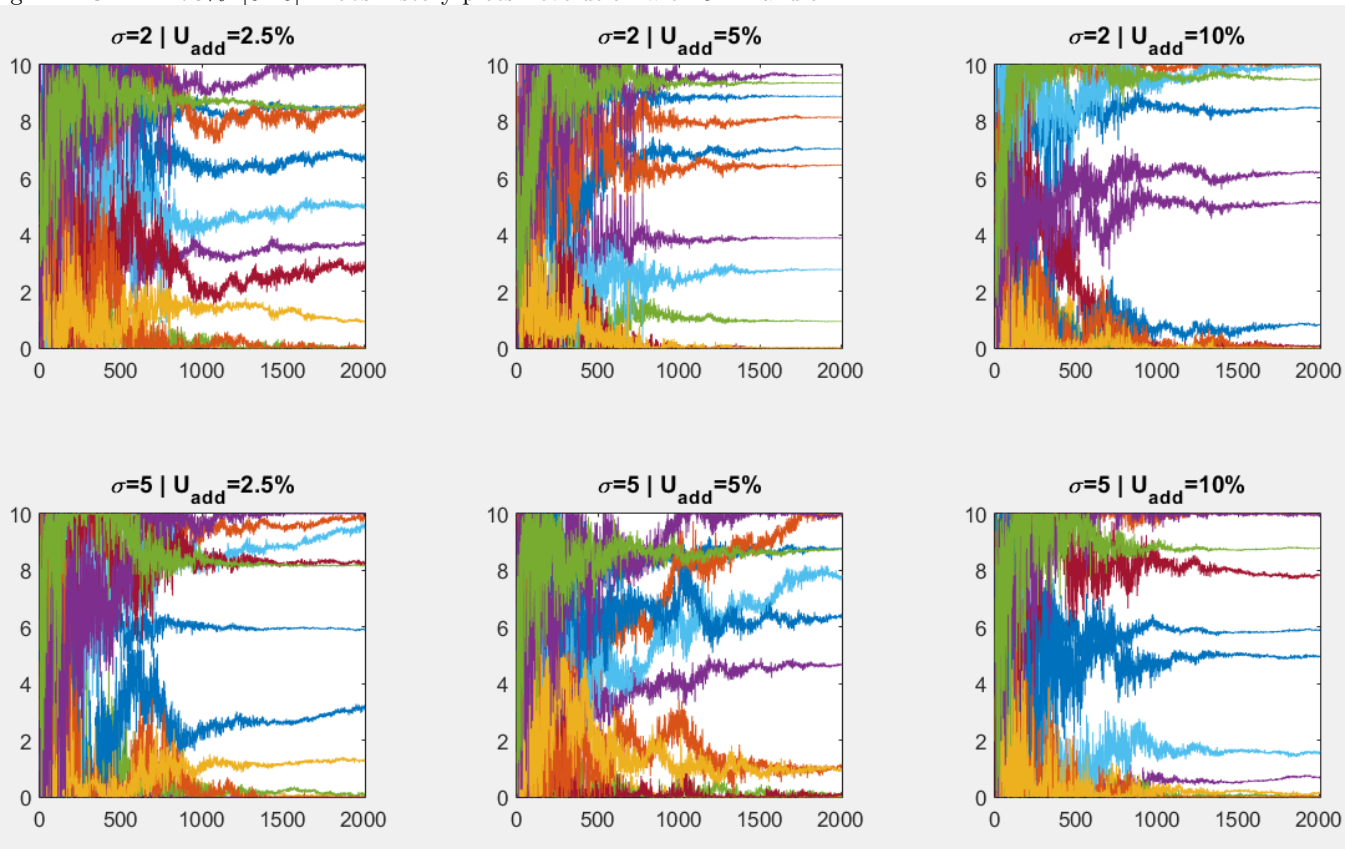


Fig. 21. $\mathbf{U^{mul}} = \mathbf{75}\%$. [0-10] knobs history plots : evolution with U^{add} and σ.



$U^{mul} = 75\%$: Fig. 20 is similar to the preceding Fig. 18. Unusual but acceptable congestion shapes that we did not observe before can appear when U^{add} is small and U^{mul} is large.
There are many situations in Fig. 21. where many knobs (up to 9 of 12) converge far from their boundaries : this is a desirable situation and is expected, because we loosened the boundaries.

Fig. 22. $\mathbf{U^{mul} = 100\%}$. Congestion pattern matching plots : evolution with U^{add} and σ.

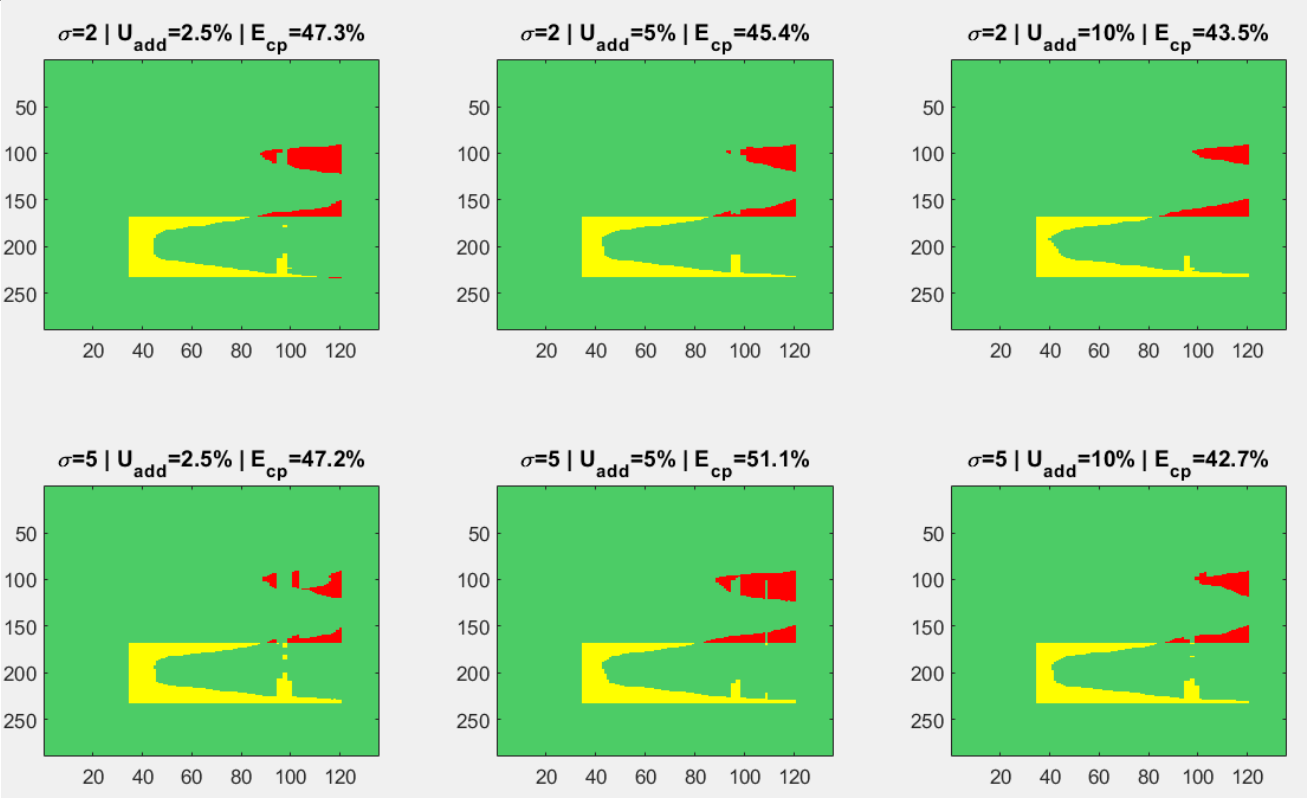
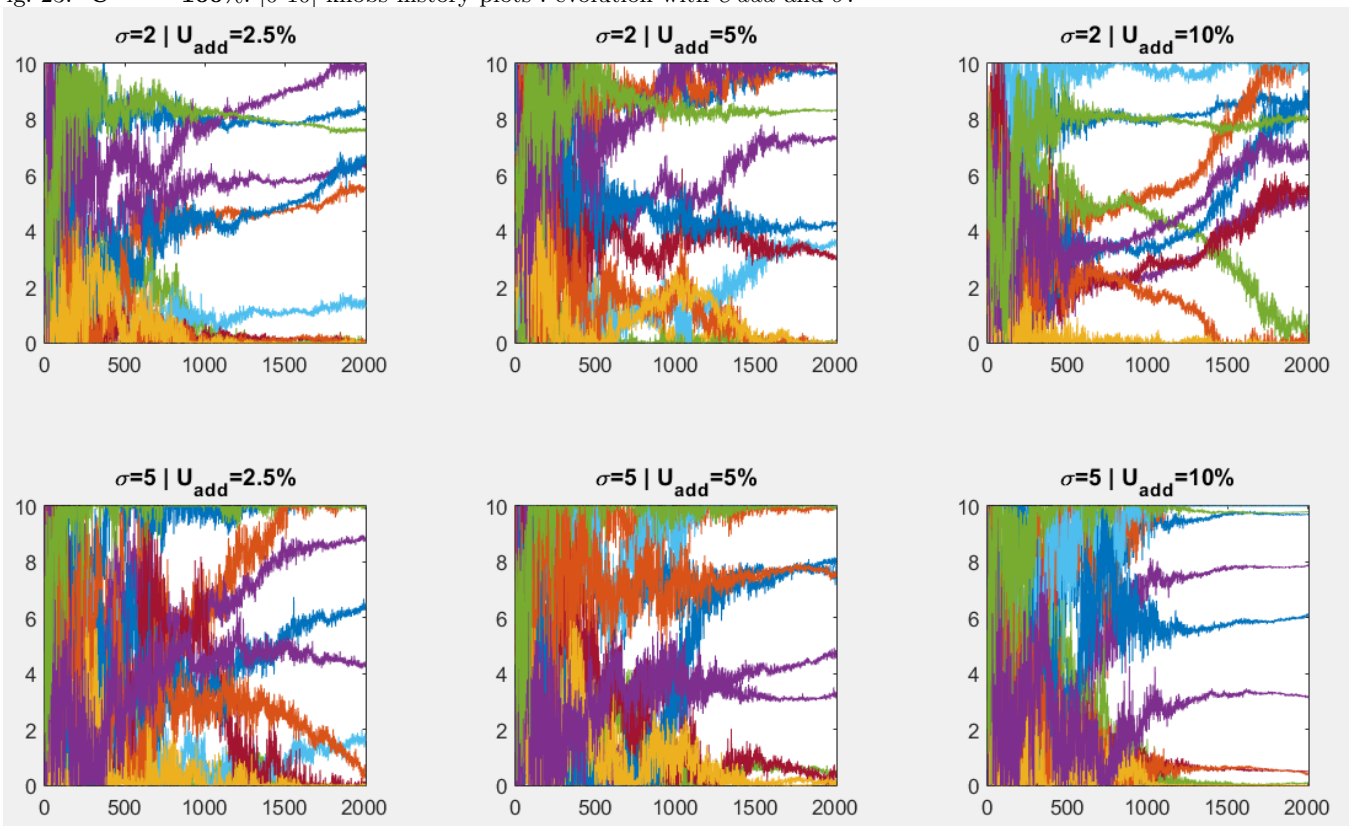


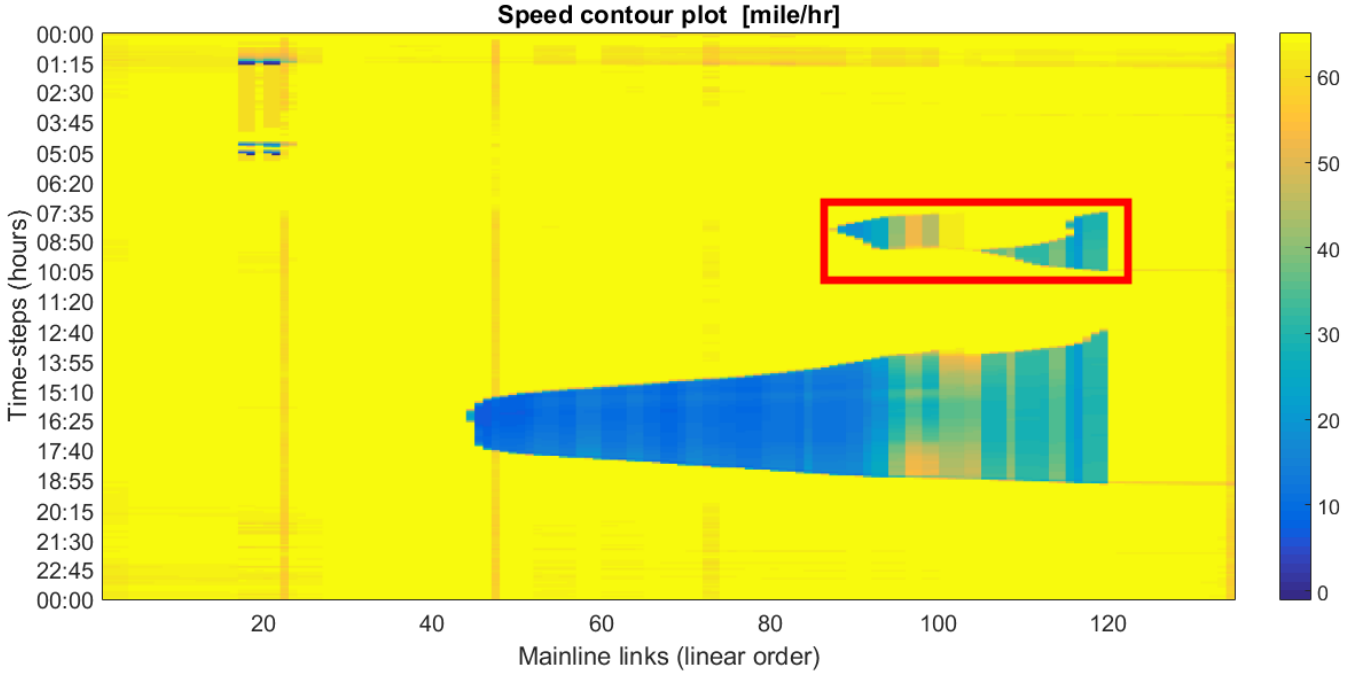
Fig. 23. $\mathbf{U^{mul} = 100\%}$. [0-10] knobs history plots : evolution with $Uadd$ and σ.



$U^{mul} = 100\%$: Fig. 22 shows too loose constraints lead too unlikely congestion situations. In this case, all the knob boundaries are their physical boundaries from $\mathscr{B}$ or are close to them. CMA-ES is not "guided" by boundaries that reflect likely traffic, it therefore finds a way of decreasing the congestion error by making non-physic "wholes" in the congestion patterns.
As we began to observe in Fig. 20, the worst cases happen when U^{add} is small. In this cases (first and second column), 6 of the knobs boundaries are very loose, and 6 are very tight : these are associated with small knob group flow demands, which always cause very tight boundaries with the multiplicative uncertainty.
Increasing U^{add} diminishes this bad effect although it does not disappear : $U^{mul} = 100\%$ is not a good choice. This effect is not caused by the congestion threshold d_i^* choice : full density or speed contour plots show the same non-physical undesirable congestion shapes. We see them in Fig. 24, a speed contour plot that intuitively reflects the congestion (located where the speeds are clearly slower). We see in this figure that the traffic does not go smoothly from congested to non-congested, implying that the framed strange shape is not due to d_i^* but to truly non-physic phenomena output by BeATS.

The knobs in Fig. 23 do not behave as expected : most of them converge to their boundaries in all the cases, when we expected them to do the opposite, as we loosened the constraints. This illustrates that CMA-ES took advantage of the new non-realistic domains of the feasible space (i.e. far from the exact equation 4), close to its boundaries (many knobs converged to 0 !). There is therefore no advantage on setting U^{mul} too large in terms of likelyhood of the resulting knob values.

Fig. 24. Speed contour plot. Example of non-physical effects on the congestion that appear when U^{mul} is too large (100 %)



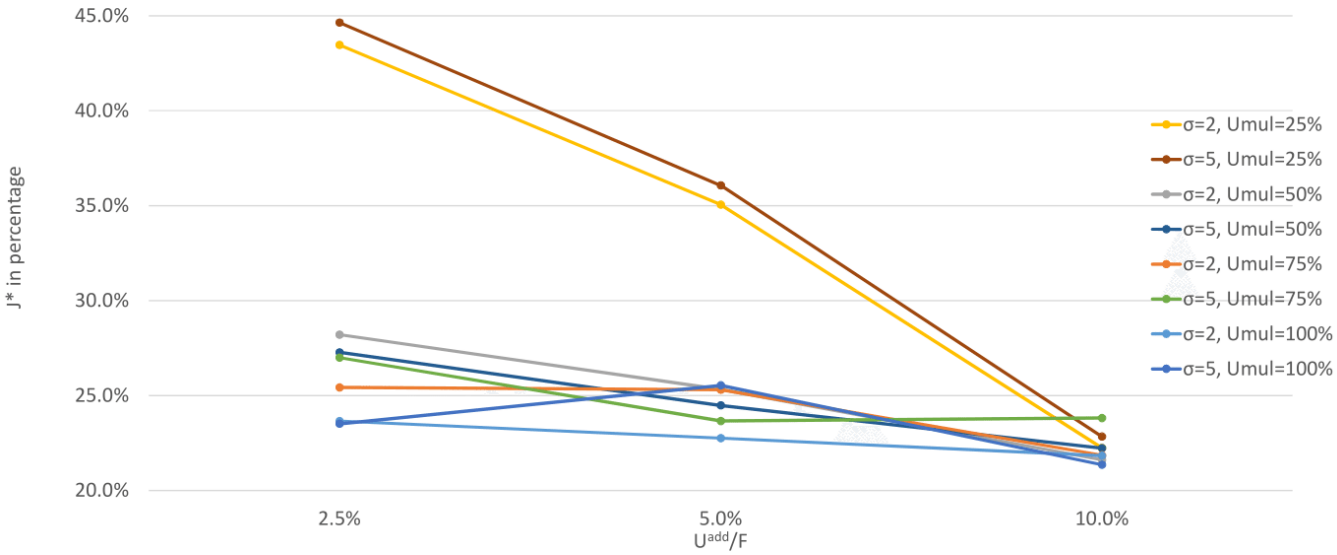
Conclusion: Our experiments showed that there is a central domain for U^{mul}, at least as wide as $[50\%, 75\%]$ in our case, that gives results of equivalent quality, the maximum allowed by the current templates shape. On the contrary, setting U^{mul} to too small values, at least $U^{mul} \leq 25\%$ in our case, prevents the algorithm to grow the congestion satisfactorily. Finally, giving too much space to the knobs, at most $U^{mul} \geq 100\%$ in our case, leads to unfeasible congestion shapes and non-realistic extremal knobs, the algorithm taking advantage of the model/simulator imperfections.

D. Changing U^{add}

For the following experiments, $\lambda = 11$.

Fig. 25 below shows the evolution of the value of J^* with U_{add}.

Fig. 25. Evolution of J^* with U^{add}



As intuitively guessed, the more uncertainty i.e. freedom to the knobs there is, the smaller J^* is. More precisely, the tendencies observed are the following :

- The only case where U^{add} drastically changes J^* is when it compensates the fact that U^{mul} is very small (25% in our case). In this situation, U^{add} always gives more freedom to the knobs than U^{mul}, and J^* improves proportionally to the increase of U^{add}. We described this effect in the last paragraph VI-C and discarded these U^{mul} too small values.
  However, this is a good way of monitoring the usefulness of U^{add}.
- For larger U^{mul} values, U^{add} helps diminishing J^* value by $2-10\%$ (depending on the relative importance of U^{add} in comparison to U^{mul}), as expected.

Below are 3 groups of figures. They display especially, for each of the 3 values of U^{add}, how the resulting congestion pattern fitting and the history of the knobs trough the execution evolve with U^{mul} and σ.

Fig. 26. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{2.5}\%$. Congestion pattern matching plots : evolution with U_{mul} and σ.

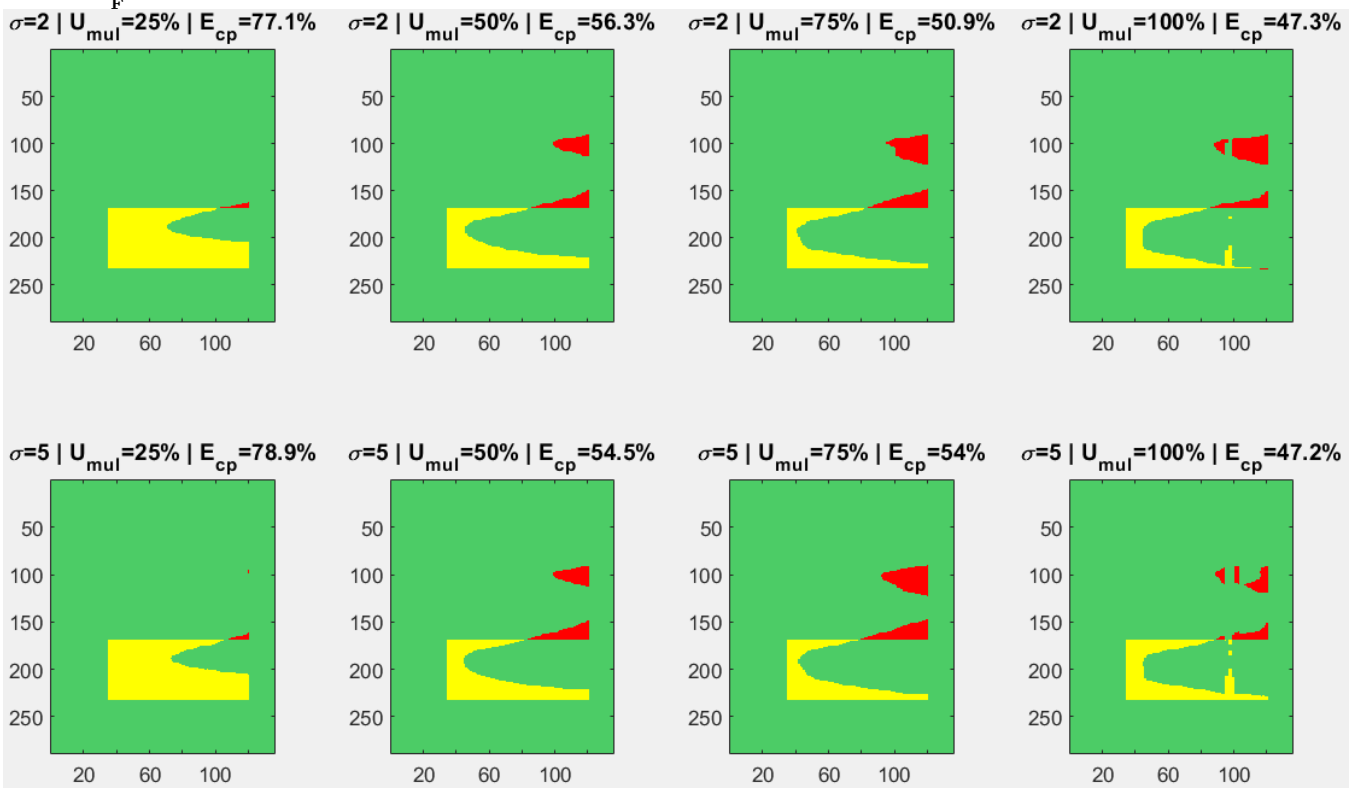


Fig. 27. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{2.5}\%$. [0-10] knobs history plots : evolution with U_{mul} and σ.

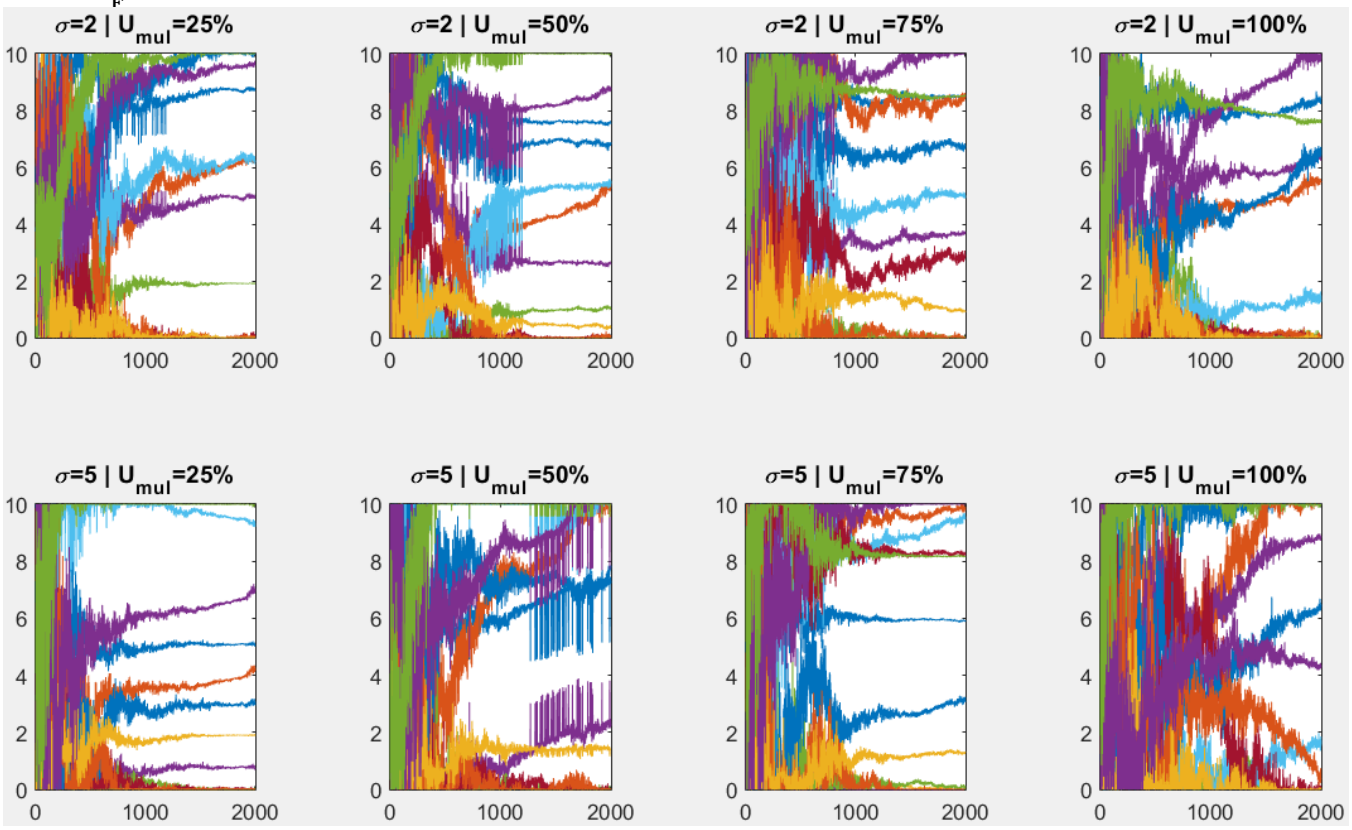


Fig. 28. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{5}\%$. Congestion pattern matching plots : evolution with U_{mul} and σ.

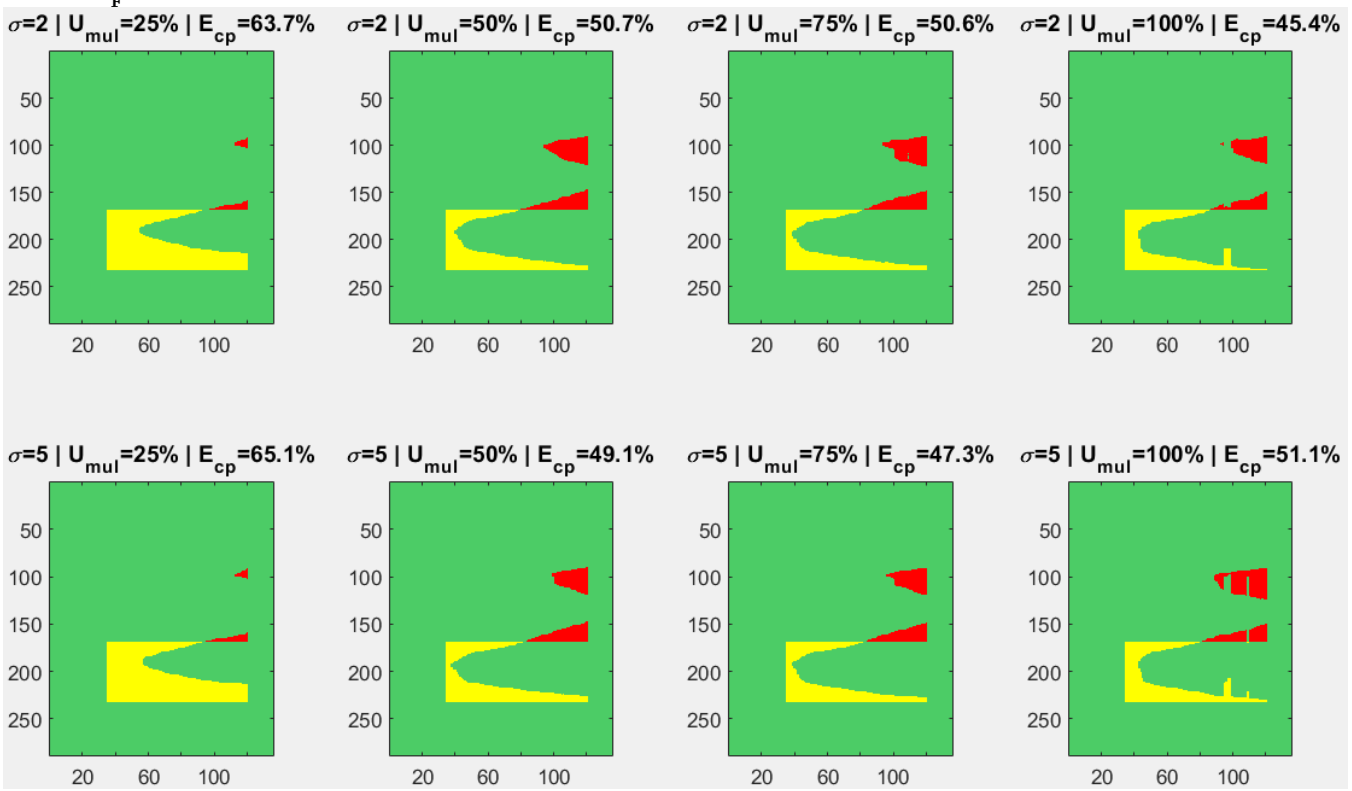


Fig. 29. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{5}\%$. [0-10] knobs history plots : evolution with U_{mul} and σ.

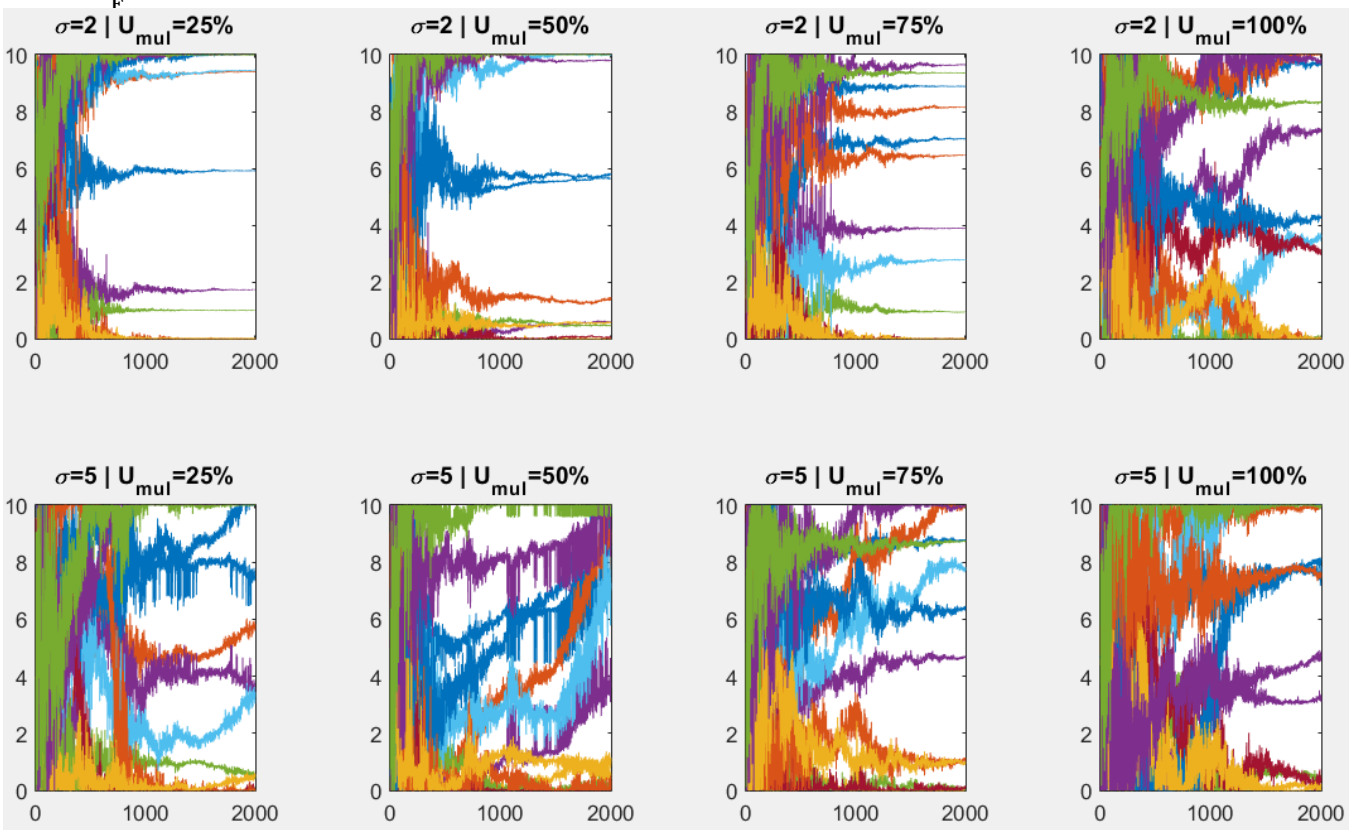


Fig. 30. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{10}\%$. Congestion pattern matching plots : evolution with U_{mul} and σ.

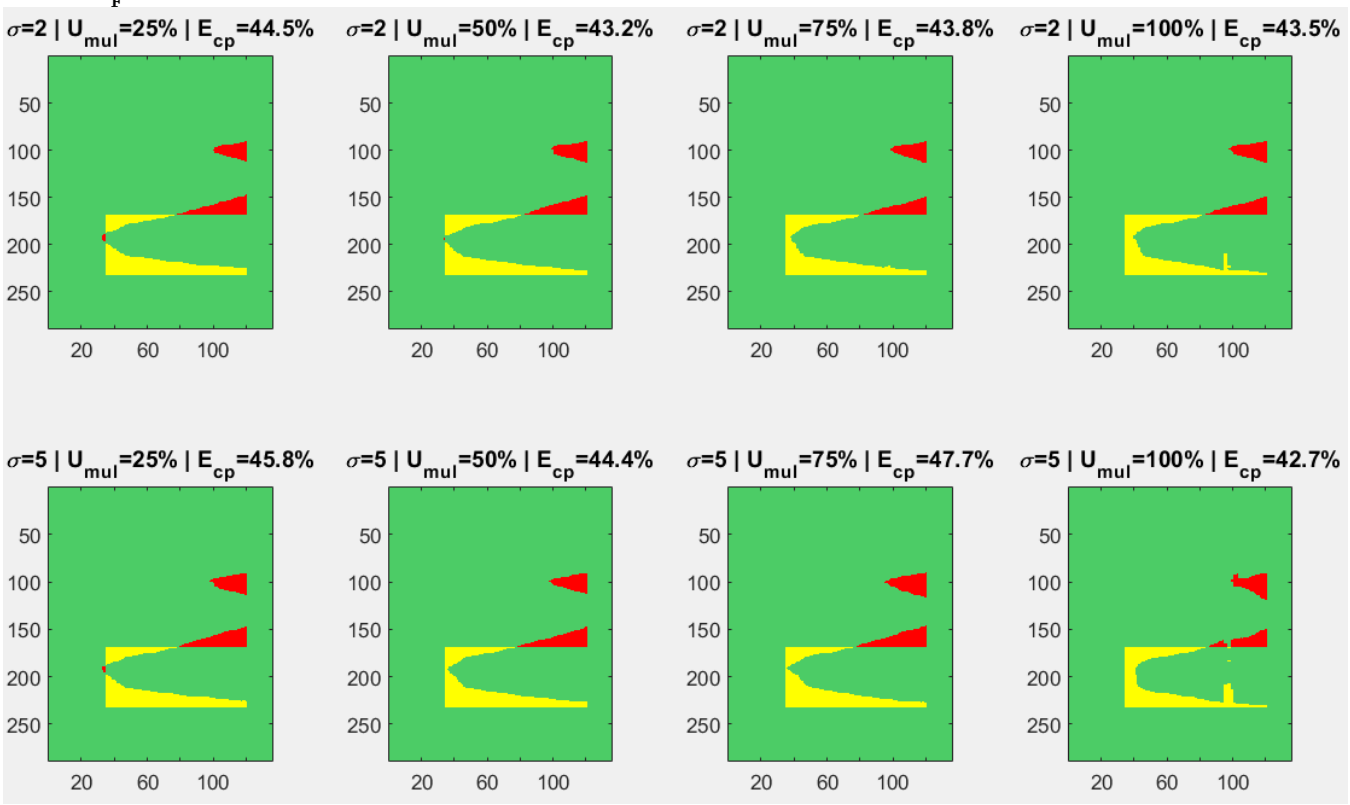
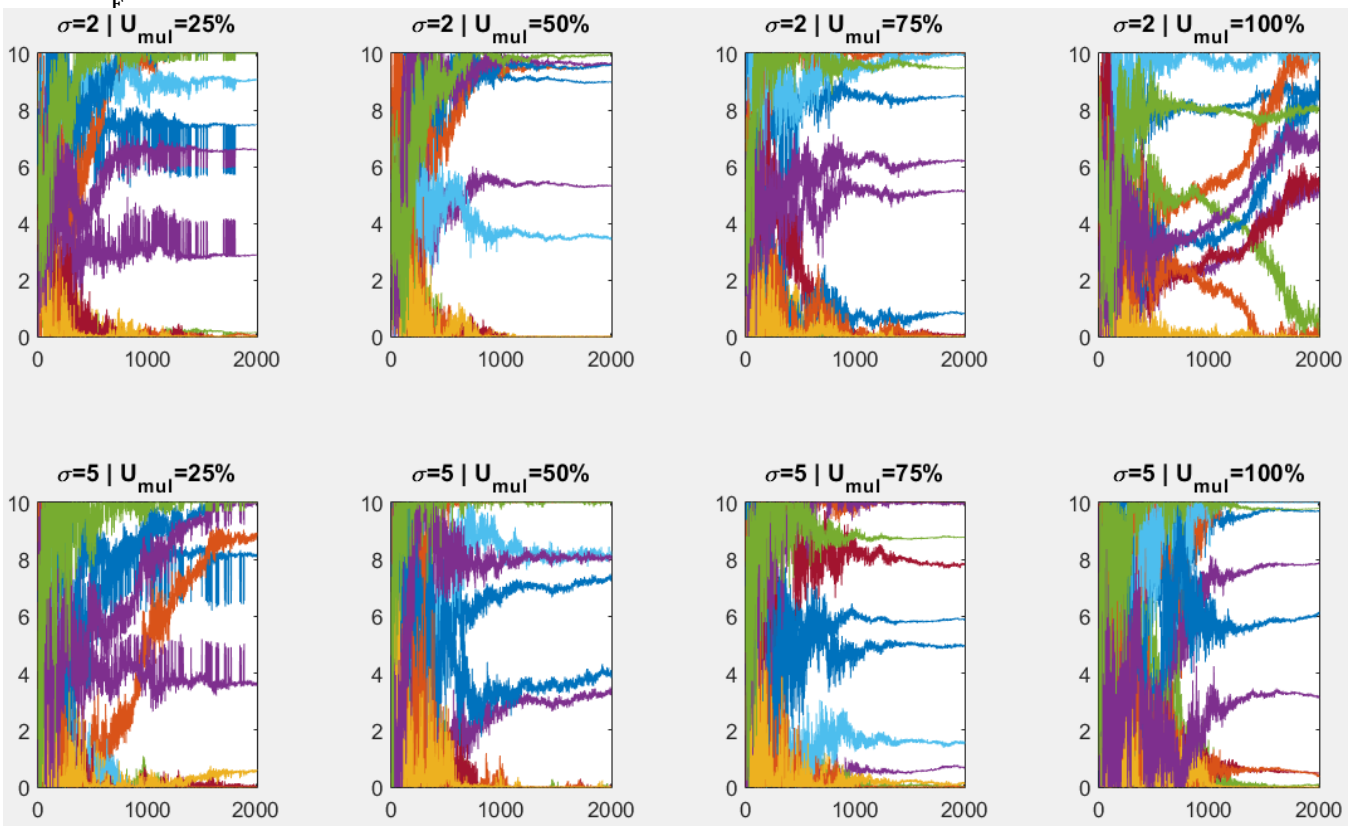


Fig. 31. $\frac{\mathbf{U_{add}}}{\mathbf{F}} = \mathbf{10}\%$. [0-10] knobs history plots : evolution with U_{mul} and σ.



The 3 preceding congestion pattern figures show how U^{add} progressively overrides U^{mul}: with $U^{add} = 2.5\%$, the resulting congestion shape is very dependent of U^{mul} (displaying the evolution described in VI-C).
With $U^{add} = 10\%$, U^{mul} has no incidence except when it is set to its biggest value, where it jeopardizes the result.

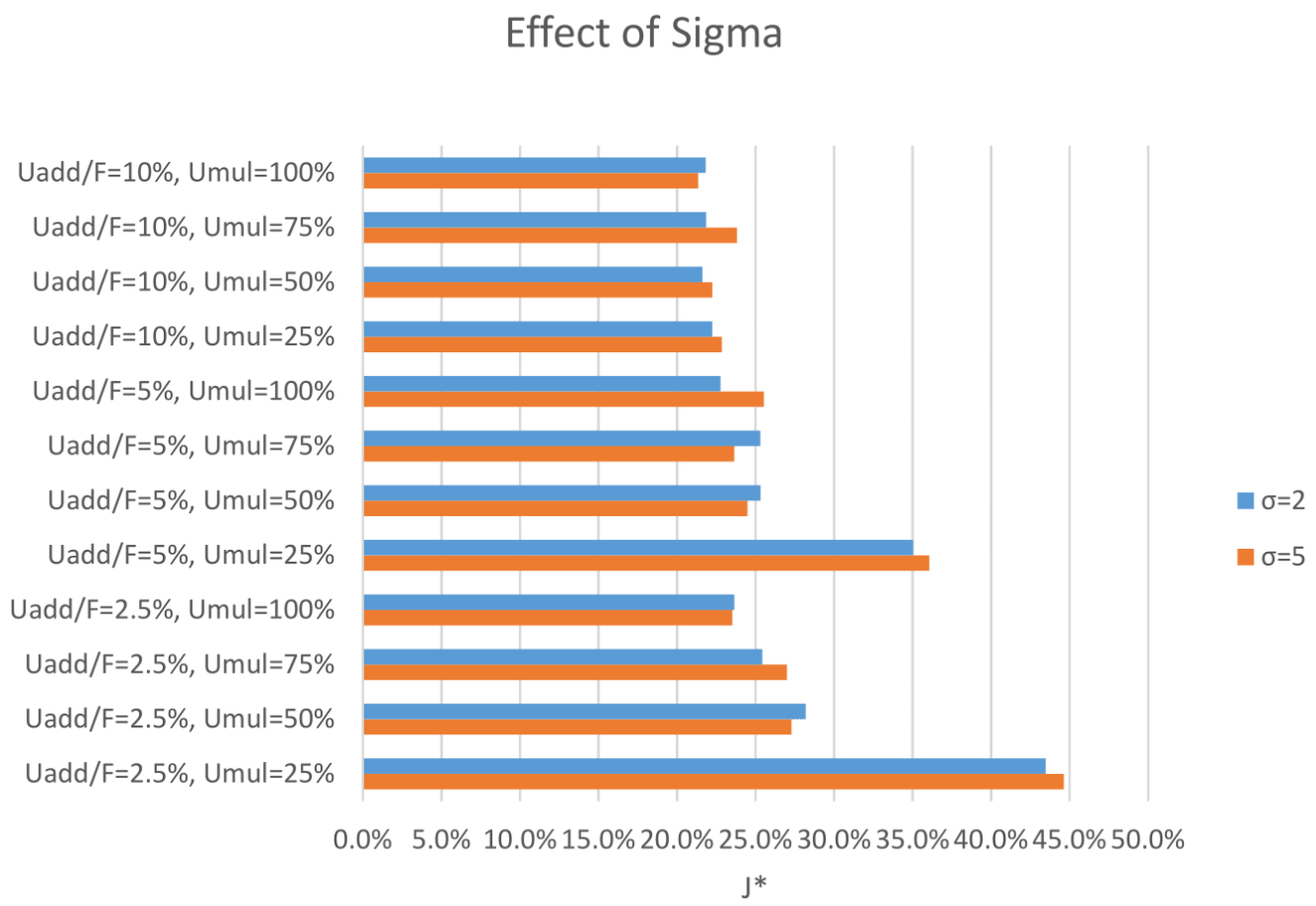
In terms of knobs evolution, the only tendency seems to be that a too large $\frac{U^{add}}{\widetilde{F}} = 10\%$ makes more of them converge to their boundaries, the algorithm taking advantage of the very wide non-realistic space it has. It is therefore preferable, if possible, to take $\frac{U^{add}}{\widetilde{F}} \leq 5\%$.

Conclusion: Increasing U^{add} tends to make uniform knob boundaries and diminish the effect of U^{mul}. Our case shows that all the values of $U^{add} \geq 2.5\%$ give the maximum "typical" result quality for the acceptable values of U^{mul} defined in VI-C.
However, U^{add} has to be set in priority in accordance with the estimated mainline sensor confidence interval.

E. Changing σ

Fig. 32 below shows the of σ on J^* with σ.

Fig. 32. Evolution of J^* with σ



σ has no influence on J^* in our case. That is because of the small search space dimension and because, due to the inexact templates shapes, many knob configurations give the same result quality (i.e. there are many equivalent (almost) global minima).

As observed in the figures of VI-C and VI-D, increasing σ makes the initial "very messy part" of the knobs convergence to be larger (up to 1000 iterations instead of 500). However, once the knobs start converging, they are more likely to stay around the same value until the end of the execution if σ is larger. This is illustrated in Fig. 23, where the comparison between the two plots of the last column shows that the knobs do not converge with $\sigma = 2$ while they do with $\sigma = 5$.
In addition, we observe in Fig. 20 that the strange congestion shapes are avoided if $\sigma = 5$, but this could be fortuitous.

Conclusion: In our case, $\sigma \in [2, 5]$ for [0-10] scaled knobs has almost no influence. There is a slight preference for $\sigma = 5$ in a few cases.
σ will probably have a greater importance if the dimension of the search space is bigger i.e. there are more knobs; or if the templates have better shapes, allowing for greater quality results i.e. fewer equivalent global minima.

F. Changing λ

The 9 last experiments we made have two goals : displaying the non-unicity of the global minima and finding the influence of the population size, especially in terms of solution unicity.

For the following experiments, $\sigma = 2$, $\frac{Uadd}{\widetilde{F}} = 2.5\%$, $U^{mul} = 50\%$ and we limit the number of BeATS evaluations to 3000.

Fig. 33. shows the exact same result quality for every execution with common parameters and for every λ value.

Fig. 33. Congestion pattern matching plots : evolution with λ.

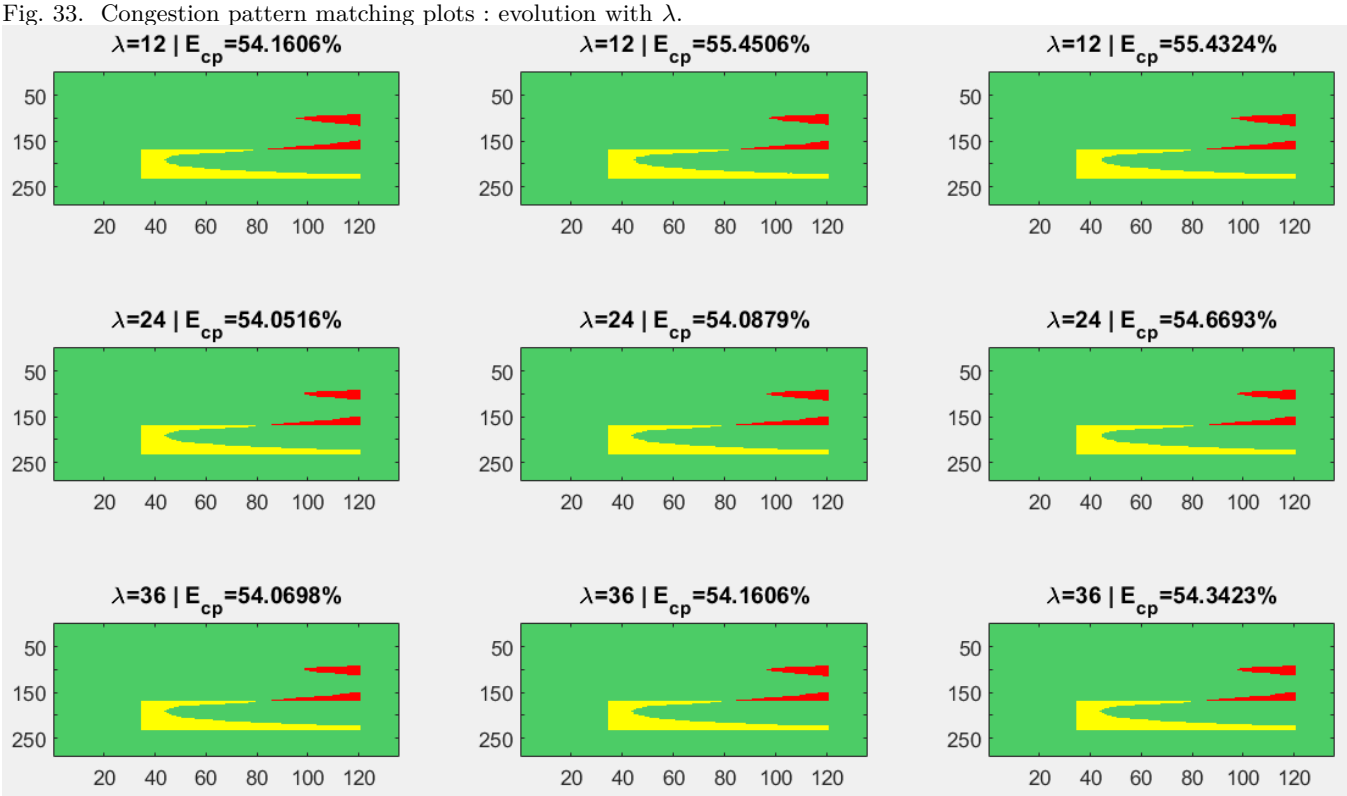


Fig. 34. shows that the knobs have very different behaviors between the executions with same λ in terms of convergence time. We also observe that the convergence time increases with λ.

Fig. 34. [0-10] knobs history plots : non-unicity and evolution with λ.

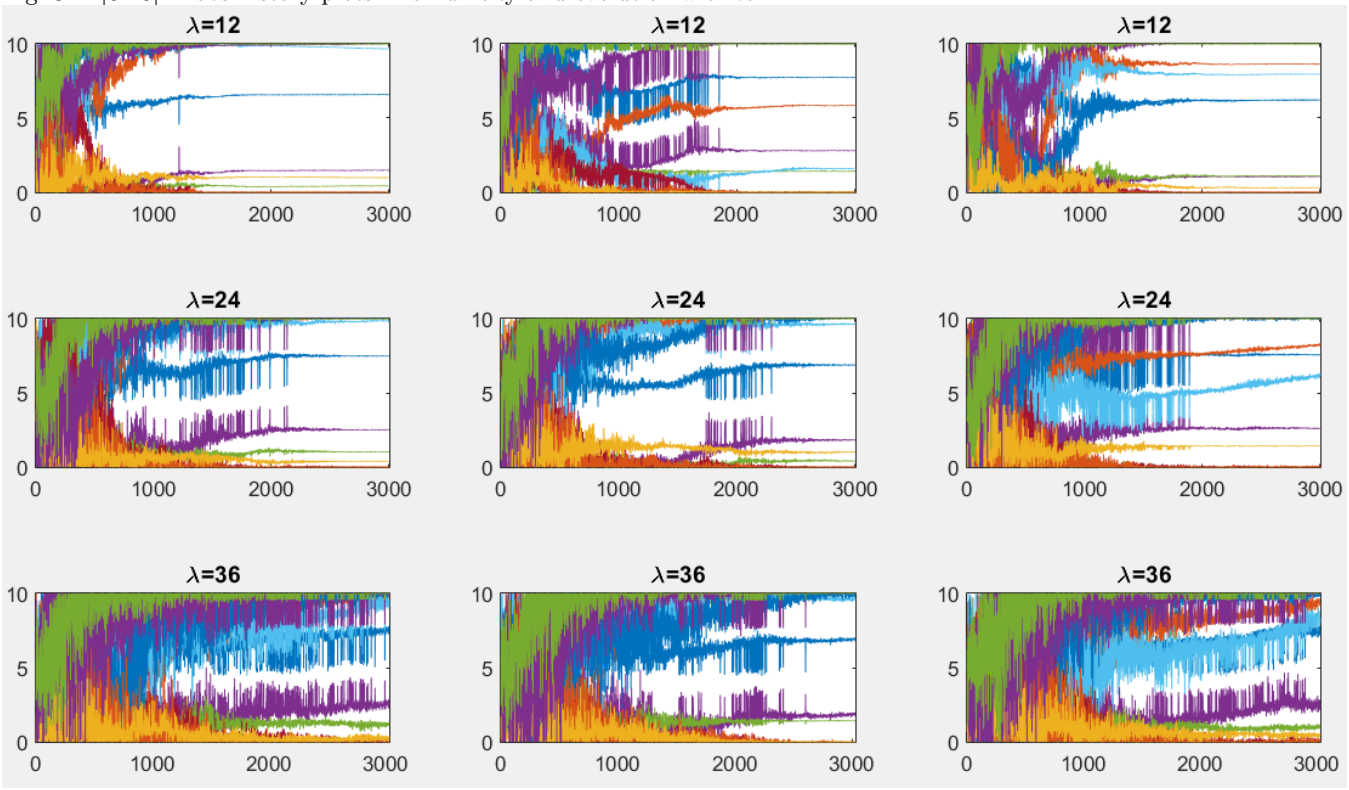
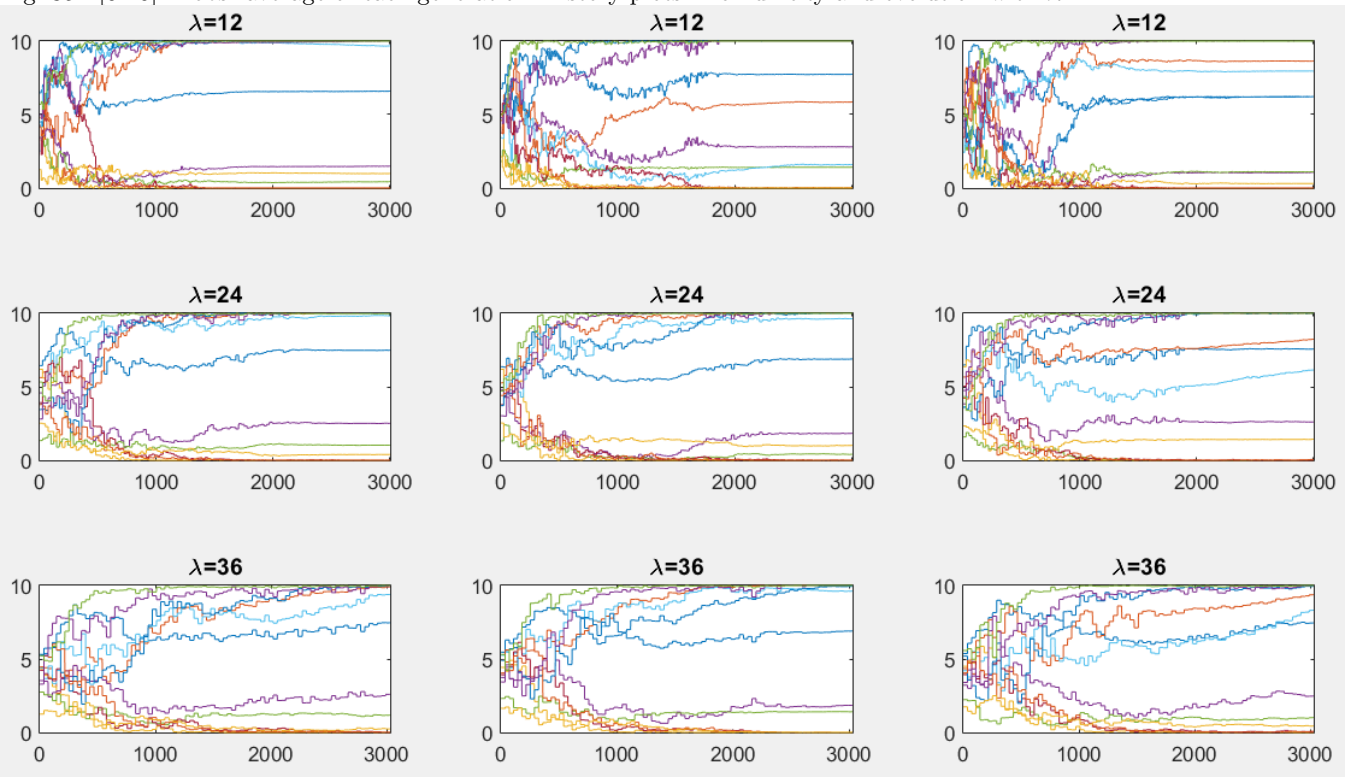


Fig. 35. which is composed by [0-10] knob plots where the values for each BeATS evaluation have been replaced by the average value of their generation. It shows that the executions with same λ converge to different knob values, whatever the λ. However, λ seems to improve slightly the solution uniqueness.

Fig. 35. [0-10] knobs "average of each generation" history plots : non-unicity and evolution with λ.



Conclusion: The experiment allows the observation of the several equivalent global minima : the knobs converge to different values but to the same result quality. Our case does not empower us to highlight any influence of λ on the result. It probably has when the search space dimension increases. It also probably has with better template shapes: in this case, there is a better discrimination between the "equivalent global minima", and increasing λ is the best way to improve global minimum search.

G. Further work

Below are the next steps to improve our calibration methods and ideas that we could apply to it:

- *Modulate the templates shape:* The inexact shapes of the templates greatly limits the result quality, the solution uniqueness and the depths of analysis that we can do on the parameters effects (especially λ and σ).
  To solve this issue, we associate each monitored ramp to a basis of normalized templates. The parameters of the problem will thus be the knob of each ramp *and* the weight associated to each template of the base of each knob. This will greatly improve the result as it will allow the "undesirable" congestion to disappear, while having credible input flow profile shapes (a combination of credible flow profiles is a credible flow profile). For example, with a basis of 3 templates, the dimension of the search space is 36 in our case, which is perfectly manageable by CMA-ES ($3 \in [3, 100]$)!
- *Better knob boundaries:* Instead of having an uncertainty approach, we could choose custom boundaries for each knob, deduced from the observation of the usual traffic around them and the bias the nearby sensor have. We could also study the effect of parameter sensitivity on CMA-ES in order to better the choose the rescale of the knob (which is today the same [0-10] scale for all).
- *Find new constraints or objectives:* The multiplicity of solution shows that the search space is too large or that we lack of information. Implementing new ways of reflecting the traffic reality on the model is how we will get solution uniqueness.
- *Use multi-objective CMA-ES:* Instead of seeing our error joint minimization problem as single-objective, we could implement the calibration as a multi-objective problem: one for each of the 3 performance errors defined in III-D. A method is exposed in [5] to use CMA-ES in multi-objective optimization.
- In terms of implementation, the parallelization allowed by CMA-ES (up to λ processes at the same time!) allows to divide the search time by λ.

As said in the introduction, this method will be part of a wider loop that will calibrate the fundamental diagrams jointly with the input flows.

VII. Conclusion

Calibration is one of the main concerns regarding the viability of macroscopic large traffic models. It consists in matching the reality of traffic with credible -if possible real- data, in order to be able to realize further experiments with the model (e.g. predict and quantify the decongestion effect of adding an off-ramp at some point of a freeway). In this paper, we expose a method to calibrate the missing input total daily flows, given their shape as a *"template"*.

To do so, we first formalize the calibration process as a general black-box optimization problem. We then choose a numerical method that is efficient and relevant for any problem : the powerful CMA evolution strategy.
We apply this numerical method to our particular freeway, large traffic simulator and template choice. The experiments show tendencies with the variation of the parameters (mainly the uncertainty). However, the inaccurate template choice, that greatly limits the best result quality, does not allow to describe deeply and accurately the effects of the parameters, especially the initial standard deviation and population size.

This paper is a report on the advancement of a work that is much wider. It will allow to calibrate jointly all the parameters of large traffic models and will be applicable to any of such, with any scenario.

Acknowledgment

This work is supported by the California Department of Transportation (Caltrans) under the California PATH program.

References

[1] N. Hansen. The cma evolution strategy. [Online]. Available: https://www.lri.fr/ hansen/cmaesintro.html

[2] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Proceedings of the 1996 IEEE Intern. Conf. on Evolutionnary Computation (ICEC '96): 312-317*, 1996.

[3] N. Hansen. (2011, June) The cma evolution strategy: A tutorial. Inria. [Online]. Available: https://www.lri.fr/ hansen/cmatutorial.pdf

[4] C. F. Daganzo, "The cell transmission model. part i: A simple dynamic representation of highway traffic." July 1993, california PATH program.

[5] N. H. Christian Igel and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *MIT Press, Evolutionnary Computation Volume 15, Number 1*, 2007.